The Dissertation Committee for Constantine Chrysostomou Certifies that this is the approved version of the following dissertation:

Addressing Intrinsic Challenges for Next Generation Sequencing of Immunoglobulin Repertoires

Committee:

George Georgiou, Supervisor

Brent L. Iverson

Jennifer A. Maynard

Hal S. Alper

Charles B. Mullins

Addressing Intrinsic Challenges for Next Generation Sequencing of Immunoglobulin Repertoires

by

Constantine Chrysostomou, B.S.; M.S.E

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

May 2014

Dedication

This dissertation is dedicated my parents, Evanthia and Andreas, and my brother, Charles. I owe all of my accomplishments to their continual support and guidance throughout my education.

Acknowledgements

I would like to acknowledge my advisor, Professor George Georgiou, who has always been a constant source for advice both scientifically and professionally. I will always be fortunate to have had him as an advisor and his continual support. In addition to Professor Georgiou, I have also had the pleasure of learning from multiple scientists who have served as advisors to me, including Professor Brent Iverson and Dr. Scott Hunicke-Smith.

I would like to thank Dr. Karthik Veeravalli who spent the time to train me and introduce me to the lab. I am also grateful for the opportunity to learn from a large number of great scientists including Dr. Sai Reddy, Dr. Xin Ge, Dr. Mark Pogson, Dr. Gregory Ippolito, and Dr. Everett Stone.

With respect to my work in bacteriology, I am greatly appreciative of my collaboration with fellow scientists Erik Quandt and Dr. Nicholas Marshall. I gained a great deal of knowledge from working with both very talented researchers. I would also like to thank two undergraduate researchers who aided my research in redox pathways: Blagoje Djordjevic and Quoc Bao Nguyen.

In my work in immunology I am indebted to the assistance provided by Benjamin Goetz, Andrew Horton, Gabriel Wu, and Dr. Sebastian Schaetzle. However, the one person who I want to thank the most is Kam Hon Hoi. Working with him has been one of the most enjoyable experiences of my graduate studies and he was intimately involved in the work discussed in almost every chapter within this dissertation.

Finally, and most importantly, graduate school has provided me with a close network of fellow researchers and friends. I greatly appreciate their constant presence and

advice as I matured as a scientist. They have left me with unforgettable memories as a graduate student and their advice throughout the years was invaluable.

Addressing Intrinsic Challenges for Next Generation Sequencing of Immunoglobulin Repertoires

Constantine Chrysostomou, Ph.D.

The University of Texas at Austin, 2014

Supervisor: George Georgiou

Antibodies are essential molecules that help to provide immunity against a vast population of environmental pathogens. This antibody conferred protection is dependent upon genetic diversification mechanisms that produce an impressive repertoire of lymphocytes expressing unique B-cell receptors. The advent of high throughput sequencing has enabled researchers to sequence populations of B-cell receptors at an unprecedented depth. Such investigations can be used to expand our understanding of mechanistic processes governing adaptive immunity, characterization of immunity related disorders, and the discovery of antibodies specific to antigens of interest. However, next generation sequencing of immunological repertoires is not without its challenges. For example, it is especially difficult to identify biologically relevant features within large datasets. Additionally, within the immunology community, there is a severe lack of standardized and easily accessible bioinformatics analysis pipelines. In this work, we present methods which address many of these concerns. First, we present robust statistical methods for the comparison of immunoglobulin repertoires. Specifically, we quantified the overlap between the antibody heavy chain variable domain (V_H) repertoire of antibody secreting plasma cells isolated from the bone marrow, lymph nodes, and spleen lymphoid tissues of immunized mice. Statistical analysis showed significantly more overlap between

the bone marrow and spleen VH repertoires as compared to the lymph node repertoires. Moreover, we identified and synthesized antigen-specific antibodies from the repertoire of a mouse that showed a convergence of highly frequent VH sequences in all three tissues. Second, we introduce a novel algorithm for the rapid and accurate alignment of VH sequences to their respective germline genes. Our tests show that gene assignments reported from this algorithm were more than 99% identical to assignments determined using the well-validated IMGT software, and yet the algorithm is five times faster than an IgBlast based analysis. Finally, in an effort to introduce methods for the standardization, transparency, and replication of future repertoire studies, we have built a cloud-based pipeline of bioinformatics tools specific to immunoglobulin repertoire studies. These tools provide solutions for data curation and long-term storage of immunological sequencing data in a database, annotation of sequences with biologically relevant features, and analysis of repertoire experiments.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

THE PROGRESSION OF RESEARCH IN IMMUNOLOGY

The immune system has evolved both innate and adaptive defense mechanisms against an ever-changing population of potentially lethal microorganisms such as bacteria and viruses. The innate immune system serves as the first level of defense which employs a system of genetically pre-encoded defensive elements against commonly encountered microbes. The adaptive immune system, on the other hand, responds more slowly but produces an immensely diverse repertoire of defensive elements. Moreover, the adaptive immune system can induce immunological memory thereby conferring prolonged or even lifelong immunity. Thus, upon infection, both the innate and adaptive systems work cooperatively to identify and precisely target previously encountered or unanticipated foreign pathogens.

This study of the immune response to infection has been a topic of interest for more than 200 years. Even at its onset, simple observations of humoral immunity led to dramatic improvements in health care and quality of life. Pioneers such as Edward Jenner and Louis Pasteur demonstrated that inoculating individuals with attenuated strains of the virus could provide apparently life-long immunity against the rampant small pox virus and rabies[1]. This revelation eventually led to the eradication of smallpox worldwide[2]. Discovery at the cellular level, however, progressed more slowly. It took another hundred years after Jenner's smallpox vaccination before Emil von Behring and Shibasaburo Kitasato demonstrated that resistance to pathogens was conferred by "anti-toxic" agents in the serum of immunized animals[1].

The emergence of modern biological technologies in the latter half of the 20th century drastically changed the tools and the pace of progress in understanding

immunology. Arguably one of the most important discoveries was that the "anti-toxic" agents observed in serum were in fact antibodies produced by specialized antibody secreting immune cells known as B cells. Newfound knowledge of antibodies and their ability to both bind with high specificity to antigens and recruit other immune cells has expanded the potential of future therapeutics. The field of antibody discovery has resulted in the development of hundreds of novel antibodies with potential therapeutic ability[3]–[5]. Despite such successes, there are still many aspects of the immune system that remain poorly understood and inhibit our advancements in next generation antibody development.

The adaptive immune response against pathogen challenge is highly sensitive to both signaling factors in extracellular environment and communication between multitudes of immune cell types. A better understanding of how these signals are integrated and how the diversity of circulating antibodies affect adaptive immunity is essential for both antibody discovery and vaccine development against more complex diseases such as AIDS[6], [7]. Until recently, large scale analysis of the entire antibody repertoire elicited in the course of the immune response was impractical given the standard molecular biology tools and instruments available.

Fortunately, the introduction of next generation sequencing at the beginning of the 21st century has again changed the landscape of immunological research. Whereas previous technology revealed extensive information concerning the cellular machinery that controls individual elements of adaptive immunity, next generation sequencing technology permits researchers to simultaneously study, at an unprecedented depth, the entire repertoire of humoral immunity at rest or following infection. Already, this technology has provided insights into the diversity of the immune repertoires, improved methods for the isolation of antigen-specific antibodies, and enabled proteomic analysis of circulating antibodies[8]–[12].

However, next generation sequencing has also introduced many challenges. Namely, the ability to sequence repertoires at great depth has presented researchers with the difficulty of not knowing how to utilize the plethora of data that is generated. In order to gain biological and therapeutic insights, there is an acute need for the development of experimental methodologies and informatics tools for acquisition and interpretation of antibody repertoire sequence data. The work described in this dissertation seeks to address this need.

FEATURES OF THE ADAPTIVE IMMUNE SYSTEM

Lymphocyte development

All cells which constitute the innate and adaptive immune systems are known as leukocytes or white blood cells. More specifically, the two main subsets of leukocytes comprising the adaptive immune system are known as B lymphocytes (B cells) and T lymphocytes (T cells). While, B cells will eventually give rise to antibody secreting cells known as plasma cells, their differentiation is highly dependent upon the mutual stimulation and maturation of T cells. T cell lymphocytes provide co-stimulatory signals to B cells in turn leading to the evolution and selection of B cells expressing antigen specific antibodies. Circulation and interaction between lymphocytes is facilitated by a specialized circulatory system known as the lymphatic system. Together, it has been estimated that there are approximately $4x10^{11}$ circulating B and T lymphocytes in humans[13]. Each mature lymphocyte expresses a highly variable B cell receptor (BCR) or T cell receptor (TCR) designed to bind foreign antigen. The number and clonal diversity of this vast repertoire of lymphocytes encoding for unique immune receptors influences an individual's response to pathogens.

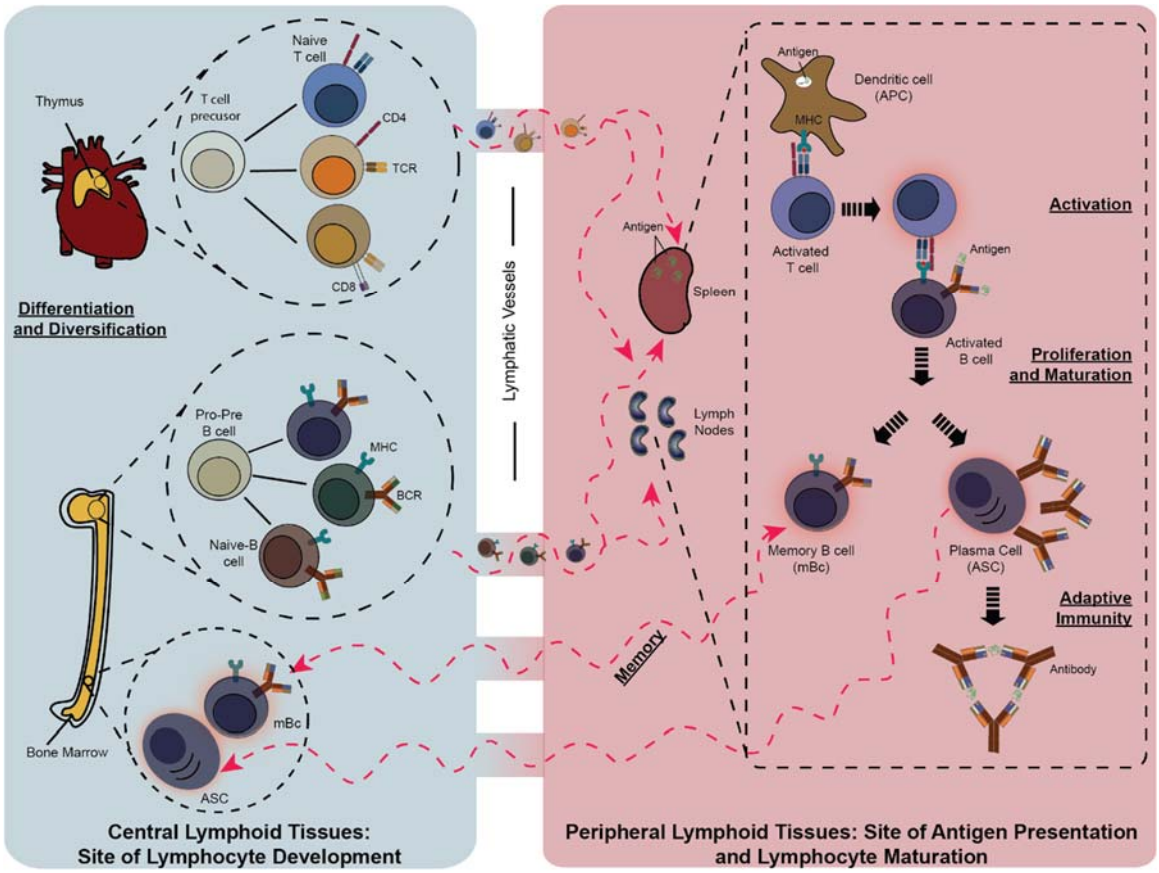


Figure 1.1: Development of lymphocytes in the lymphatic system

B and T cell lymphocytes differentiate from precursor cells in the bone marrow and thymus lymphoid tissues, respectively. Differentiated naïve cells exit central lymphoid tissues via specialized lymphatic vessels and enter peripheral lymphoid tissues where they survey for antigen challenge. Stimulation between activated T and B cells results in B cell proliferation and maturation. Matured B cells can circulate back and take residence in the bone marrow.

B and T cells originate in the bone marrow and thymus, respectively. During development in the central, or primary, lymphoid tissues, lymphocytes undergo similar differentiation pathways in which the genes encoding the BCR and TCR undergo various diversification processes. Differentiated cells expressing functional yet non self-reactive

BCR and TCR are termed naïve B cells and naïve T cells, respectively. Following differentiation, the primary repertoire of naïve T cells and B cells egress from their respective compartments, the thymus and bone marrow, and circulate to peripheral lymphoid tissues where antigen specific lymphocytes can become activated and proliferate (Figure 1.1).

Foreign substances and microorganisms drain into peripheral, or secondary, lymphoid tissues via lymphatic vessels. Secondary lymphoid tissues, which include the lymph nodes, spleen, and mucosal tissues, are composed of densely organized layers of leukocytes that survey for pathogenic species present in lymph. Interactions between cells from both the innate and adaptive immune system in secondary lymphoid tissues are required for an effective immune response. For example, in secondary lymphoid tissues, antigen presenting cells of the innate immune system, known as dendritic cells, display epitopes from engulfed microorganisms on their cell surface[14]. T cells expressing antigen specific TCR can be activated by antigen presented on dendritic cell surface receptors and differentiate into either killer T cells, regulatory T cells, or helper T cells. Activated helper T cells are responsible for activating the maturation of antigen specific B cells in peripheral lymphoid tissues.

B cells that bind antigen while receiving proliferative co-stimulatory signals from helper T cells can further develop into long-lived memory B cells or antibody-secreting cells with either a transient or protracted lifespan. Memory cells survive for very long times and provide immunological memory, ready to proliferate rapidly in response to re-challenge with pathogen. Mature B cells such as plasma cells and plasma blasts produce and secrete copious amounts of the soluble form of the BCR which corresponds to the antibodies found in secretions and in circulation. Antibodies secreted into blood are tasked

with both binding to foreign objects with high specificity, and with initiating an effector response against the target by interacting with leukocytes of the innate immune response.

Antibody structure and function

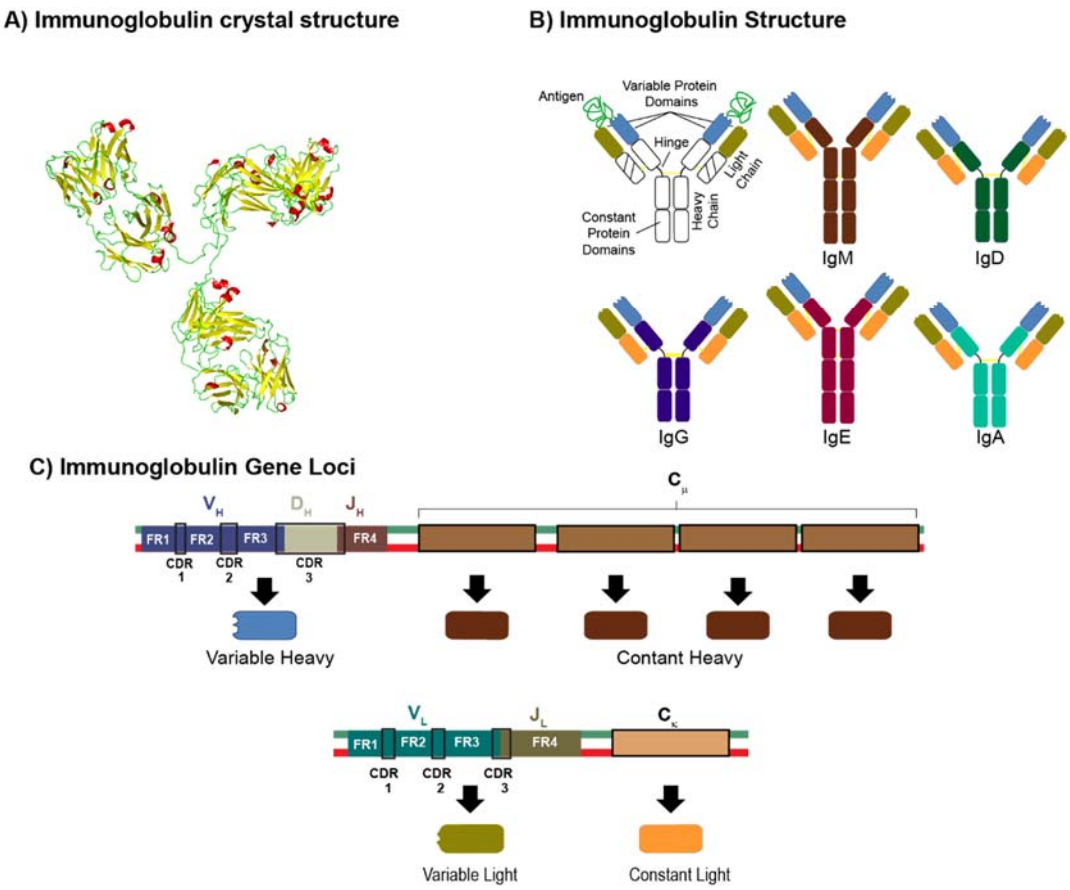


Figure 1.2: Structure of antibodies

A) Crystal structure of an antibody (PDB: 1IGT) B) Cartoon representation of antibody structure. The N-terminal "Y-shaped arms" of the antibody bind antigen. The heavy chain encodes for the C-terminal "stem" responsible for effector function. Antibodies express five main classes of constant chains. C) Both heavy chain and light chains are composed of a variable region and constant region. The heavy chain variable region is encoded by three gene segments, V, D, and J. The light chain variable region is encoded by a V and J gene segment.

Antibodies are described as "Y" shaped molecules containing two identical heavy (50 kDa) and light (25 kDa) chains linked together by disulfide bonds (Figure 1.2 A&B). The "arms" of the antibody are responsible for binding antigen tightly whereas the C-terminal "stem" is responsible for determining the immune system response to the bound antigen. The presence of two identical binding sites at the N-terminus increases the effective affinity for the antigen.

Structurally, both the heavy and light chain are comprised of multiple homologous protein domains. The overall conformation of each domain is known as an immunoglobulin fold[15]. The immunoglobulin folds contain a canonical secondary structure formed from two beta sheets linked via disulfide bond; anti-parallel beta strands that constitute the beta sheets are connected together by flexible loops[16]. The N-terminal domains of each chain are known as the variable heavy (V_H) and variable light (V_L) domains. The remaining domains comprise the constant regions. Heavy chain antibodies contain between four and five immunoglobulin domains (C_H) whereas the light chain is composed of two such domains (C_L) (Figure 1.2B).

In humans, the V_H region of an antibody can be associated with any of the five C_H domains encoded by the separate μ, δ, γ, ε, and α heavy chain gene clusters (Figure 1.2 A)[5]. Each C_H class confers different characteristics to an antibody such as antibody half-life, effector functions (i.e. ability to recruit innate cells against the bound pathogen), and location within the body; these characteristics determine the antibody's isotype. For example, naïve B cells egressing from the bone marrow will express two classes of B cell receptors, IgM and IgD, encoded by the μ and δ gene C_H clusters, respectively. IgM expressing cells can form pentamers, significantly increasing the overall avidity of naïve cells with low affinity for the antigen[5]. Furthermore, once a naïve cell matures into highly specific antibody secreting cells, a process known as class-switch recombination

(CSR) swaps the μ C_H gene cluster for a γ, ε, or α C_H gene cluster[17]. Most affinity matured antibodies present in the blood and extracellular fluid belong to the IgG isotype and thus express a γ C_H constant region. The IgG class of antibodies is highly efficient at recruiting innate leukocytes to eliminate bound antigen via antibody dependent cell phagocytosis (ADCP) or cytotoxicity (ADCC)[18]. IgA and IgE isotypes are predominantly responsible for countering pathogens present in the mucosal and epithelial layers[19], [20].

While the constant heavy chain determines antibody isotype, the variable heavy (V_H) and light (V_L) domains are responsible for antigen binding. Specifically, the V_H region is encoded by three gene segments, a variable (V), diversity (D), and a joining (J) gene segment, and the V_L region is encoded by only a germline V and J gene segment (Figure 1.2 C)[21]. Sequence variability in the V_H and V_L is not evenly distributed, but concentrated in three hypervariable regions called the complementary determining regions (CDRs 1, 2, and 3)[21], [22]. CDRs 1 and 2 are found in the variable gene segment whereas CDR3 is formed by the junctions of the V-D-J and V-J gene segments comprising the V_H and V_L, respectively[23]. The CDRs correspond to the flexible loops between anti-parallel beta strands of the immunoglobulin fold and provide the vast majority of contact with antigen[16]. The less variable protein sequences which flank the CDR regions are called the framework regions (FR 1, 2, 3, and 4)[21].

Genetic diversification processes that guide repertoire diversity

The true diversity of an individual's immune repertoire is currently unknown. Although there are more than 10^{11} circulating lymphocytes in humans, genetic diversification processes introduced during the development of a pro-B cell into an affinity matured B cell can theoretically produce a significantly larger number of variants

expressing unique surface receptors. Specifically, the diversity of an antibody is regulated by four genetic mechanisms: (1) V(D)J recombination, (2) N/P nucleotide addition between the V(D)J junctions, (3) paring of heavy and light chains, and (4) somatic hypermutation introduced during affinity maturation of antigen specific lymphocytes in germinal centers[21].

V(D)J Recombination

The gene segments which encode heavy and light chains are organized into three separate chromosomal clusters. In humans, the V_H, D_H, and J_H gene segments are found in chromosome 14[24]. While there is only one genetic locus for the heavy chain, two independent loci, located on chromosomes 22 and 2, give rise to lamda (λ) and kappa (κ) light chains[25], [26]. In precursor lymphocyte cells, these germline encoded loci contain multiple, non-identical copies of each V, D, and J gene segment[27]. Specifically, 140 unique V_H genes, 27 unique D_H genes, and 6 unique J_H genes have been identified in the V_H locus of chromosome 14 in humans; similarly, the lamda and kappa genetic loci of human precursor cells encode for 75 V_λ and 82 V_κ genes, and 7 unique J_λ and 5 J_κ genes[28]. During lymphocyte differentiation, the supposed random recombination of these germline gene segments, termed V(D)J recombination, can produce a highly diverse, $>10^7$, population of naïve lymphocytes.

V(D)J somatic recombination is initiated by lymphocytes undergoing differentiation in primary lymphoid tissues. During this process, one $V_{H[\lambda/\kappa]}$, one D_H, and one $J_{H[\lambda/\kappa]}$ germline gene are recombined into a single V_H or V_L exon[21]. Briefly, lymphocyte precursor cells express two recombinase enzymes, known as RAG-1 and RAG-2, that catalyze the random cleavage of DNA segments between specific signal sequences that flank V_H and V_L genes[29], [30]. In early stage B lymphocytes, the RAG-

1/2 complex directs the random selection and cleavage of chromosomal DNA between the D_H and J_H germline genes (DJ recombination) [31]. Next, a random V_H gene is recombined with the DJ gene segment in cells that have differentiated into the pre-B stage (Figure 1.3)[31].

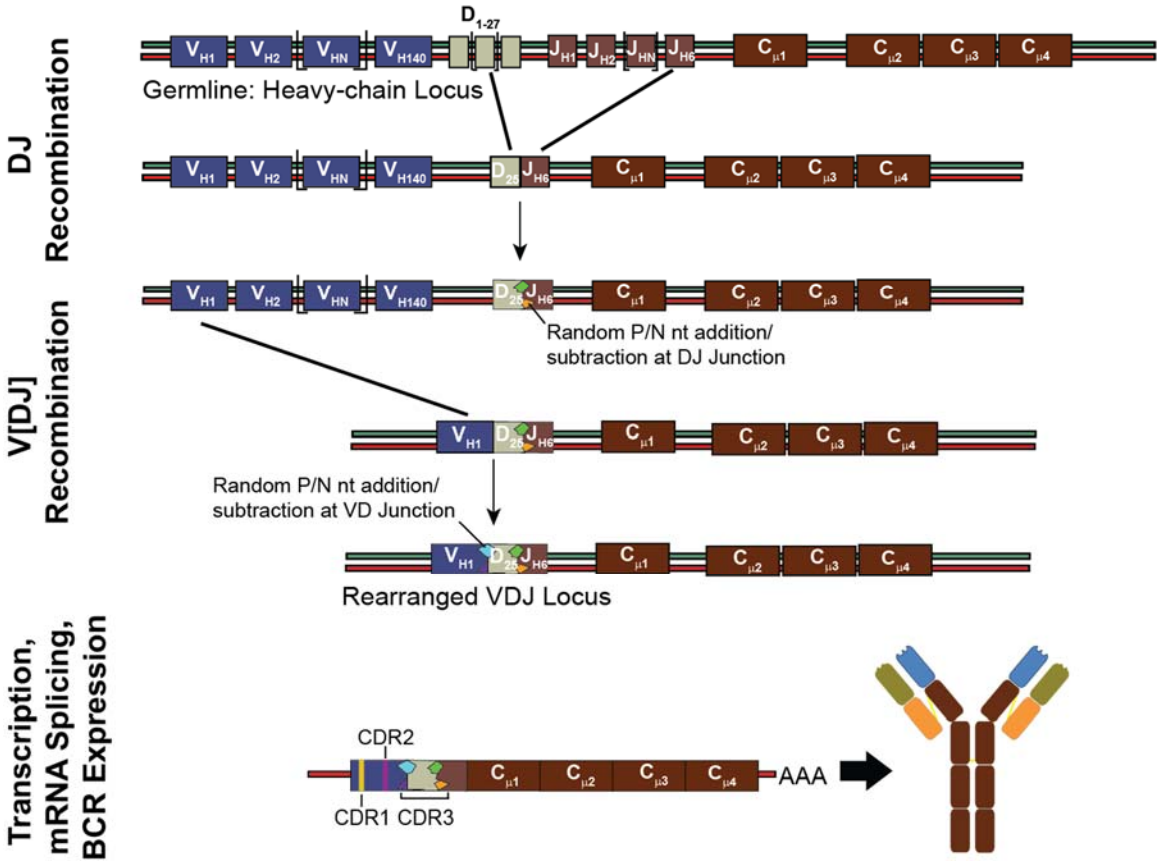


Figure 1.3 Somatic recombination of the V_H gene segment in a Pro-Pre-B cell

The germline of precursor lymphocytes encodes for multiple copies of a V_H, D_H, and J_H gene segment. During differentiation, lymphocytes select one V_H, D_H, and J_H from the germline (V(D)J recombination). First, DJ recombination occurs during differentiation into a Pro B cell. Next, differentiation into a Pre B cell results in a VDJ recombined V_H gene. During VDJ recombination, random N/P nucleotide addition and subtraction introduced at the VD and DJ junctions (illustrated by colored diamonds above) results in the formation of a highly unique region of the antibody termed the CDR3.

Identical recombination of the $V_\kappa J_\kappa$ or $V_\lambda J_\lambda$ gene segments into a V_L exon occurs during the differentiation of a pre B cell into an immature B-cell[31]. While somatic recombination of germline genes can theoretically encode for more than 10^7 combinations of unique V_H/V_L pairs, V(D)J recombination is accompanied by an additional diversification process that introduces significant sequence variation at the V(D)J junctions, as discussed in the next section.

N/P nucleotide addition contributes to CDR3 diversity

In naïve B cells that have not been exposed to antigen, the CDR1 and CDR2 complementary determining variable regions are encoded entirely by the variable V_H and $V_{[\kappa/\lambda]}$ germline gene segments. The CDR3, on the other hand, is formed by the action of DNA repair mechanisms that are recruited after V(D)J recombination. RAG-1/2 directed joining of the gene segments creates hairpins which must be resolved by the nuclease enzyme, Artemis. Resolution of these hairpins results in the random addition of palindromic nucleotide sequences (P addition) at the ends of the joining region[32]. In addition to P addition, a lymphoid-specific enzyme, terminal deoxynucleotide transferase (TdT), randomly adds nucleotides to the ends of the V(D)J gene segments resulting in random non-templated nucleotide addition and subtraction (N addition) (Figure 1.3).

Because of random nucleotide N/P addition at the V(D)J joining ends of both heavy and light chains, the CDR3 is the most unique and variable region of an antibody. For example, in humans, the length of the CDR3 sequence in the heavy chain can vary between 5 and 30 amino acids. Introduction of this random junctional diversity is the reason CDR3 sequences are often used as fingerprints to identify groups of B cells that are clonally expanded from the same parent naïve cell that encodes for a unique CDR3. Precursor B cells which express non-self-reactive and functional IgM and IgD B-cell receptors are

known as mature B cells. This repertoire of mature B cells is also considered naïve since it is not yet known which B cells can successfully engage future pathogens.

Affinity maturation in germinal centers

Naïve cells leave primary lymphoid tissues and circulate through blood and lymphatic vessels to secondary lymphoid tissues, surveying for potential antigens. Only antigen specific B cells that receive the proper signals from helper T cells and cytokines are programmed for survival and proliferation. These selected B cells can further differentiate into either short-lived plasma cells, memory B cells, or long-lived plasma cells[33]. Short-lived plasma cells , or plasmablasts, live for several days and produce low affinity antibodies early on during an immune response, whereas differentiated memory or plasma cells can express much higher affinity B cell receptors and antibodies[34], [35]. More importantly, selected memory B cells are capable of rapidly responding to repeated exposures to the pathogen. While the fate of B cell differentiation is not governed by one process, a major source of high affinity memory B cells and long-lived plasma cells is within specialized sites of peripheral lymphoid tissues called germinal centers[34], [36].

Germinal centers start as a small cluster of dividing antigen-stimulated B and T cells. Histological staining of germinal centers shows that, after a few days, dividing cells expand into two well defined compartments of light and dark stained zones[22]. These two distinct zones are believed to play a key role in affinity maturation and selection. According to the cyclic re-entry model of affinity maturation within germinal centers, B-cell clones enter the dark zone where they undergo division and another chromosomal diversification process known as somatic hypermutation (SHM)[36]. This high density of proliferating B cells gives the compartment its “dark” appearance after histological staining[37]. Somatic hypermutation is catalyzed by the enzyme, activation-induced

cytidine deaminase (AID), which randomly creates base-pair mismatches via the deamination of cystosine into uridine[38]. Introduction of these base-pair mismatches signals DNA repair pathways, but in the process also introduces semi-random nucleotide changes (approximately 10^{-3} to 10^{-5} mutations per base) at sites near the deamination site[39], [40]. After SHM, B cells encoding diversified BCRs are interspersed into the germinal light zone where they undergo selection. Clones with increased affinity for the antigen are selected for further diversification or differentiation whereas low affinity mutants are programmed for apoptosis[36] (Figure 1.4).

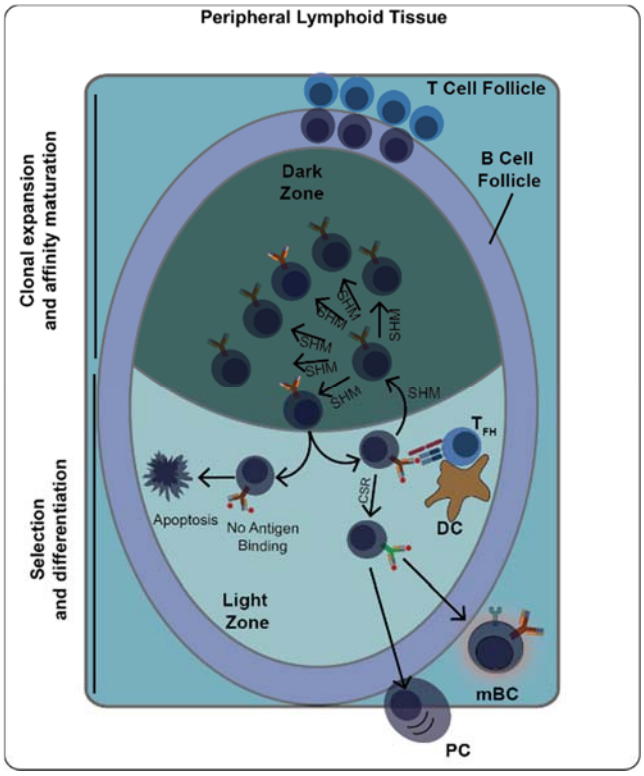


Figure 1.4 The cyclic reentry model of affinity maturation

T cell activated B cells form germinal centers (GC) in peripheral lymphoid tissues. In the dark zone of a GC, B cells undergo clonal expansion and SHM of their respective BCRs. Expanded B cells enter light zone where they are presented with antigen. BCR displaying affinity for the antigen can re-enter dark zone for further SHM. BCR with low antigen affinity are programmed for apoptosis.

In summary, studies in molecular genetics have revealed that the adaptive immune system can encode for an astronomical number of unique B cell receptors which arise as the result of multiple diversification and antigen-selection processes. During an immune response, unique B cell lymphocytes capable of recognizing the pathogen with high affinity are produced and deployed to block the infectious agent and mediate its clearance through the engagement of innate killer cells. Such mechanistic studies, however, can only provide a partial picture of adaptive immunity during infection and little is known about the true diversity and interaction of the circulating immune repertoire as a whole.

HIGH THROUGHPUT TECHNOLOGIES IN IMMUNOLOGY

Significant improvements in high-throughput technologies have allowed for "high-resolution" system level analysis of the immune receptor repertoire. The combination of cell cytometry (for the isolation of desired lymphocyte subsets), single cell analysis and/or next generation sequencing can enable both the quantification of the percentage of antigen specific lymphocytes and identification of the nucleic acid sequence of their respective BCR or secreted antibodies. Table 1.1 summarizes current technologies that have been used by immunologists for sequencing and investigating the functional analysis of antibodies and T cell receptors expressed by lymphocytes.

High Throughput Technology		Description	Cells Analyzed	Applications	Pros	Cons
	Hybridoma	B cells immortalized into tumor cells	<1	Antibody structure and antigen binding	Immortalized; DNA Ig sequence specific to one cell	Low efficiency
Single Cell Cloning	Microtiter plates	Sorted B cells individually plated into wells of microtiter plates	10^2	Antibody discovery; Single cell functional analysis	DNA Ig Sequence specific to one cell	Low throughput
Single Cell Cloning	Microengraving techniques	Plates contain > 10^5 wells	10^5	Antibody discovery; Single cell functional analysis	DNA Ig Sequence specific to one cell; High throughput screening	Expensive; Limited by single cell cloning
	Next-generation sequencing	High throughput sequencing of pooled mRNA from entire populations	10^6	Repertoire analysis; antibody discovery	Highest Throughput; simple and fast method	Sequence error; Ig Sequences not linked to specific B cells

Table 1.1: Current technologies in immunology

Comparison of the technological capabilities of hybridoma, single cell analysis, and next generation sequencing in immunology. Table adapted from review by Reddy and Georgiou, 2011.

Single Cell Analysis

Early studies of B cells and antibody discovery relied upon hybridoma technology in which individual B cells were immortalized by fusion with tumor cells[41], [42], [43], [44]. The discovery of hybridoma technology was a milestone for immunology research,

antibody discovery, and the understanding of B cell fate and regulation[45]–[47]. However, reliance upon hybridomas for in-depth studies of immune repertoires is impractical because successful formation of immortalized cells occurs at a very low frequency. More importantly, not all B cell types, such as terminally differentiated plasma cells, are amenable to immortalization[48]. The subsequent introduction of single cell cloning overcame some of these limitations. Single cell cloning involves four general steps: (1) isolation of specifically differentiated lymphocyte cell populations with the help of antibodies that recognize the appropriate cluster of differentiation (CD) cell surface protein markers using fluorescence activated cell sorting (FACS); (2) plating of cells on microtiter plates at limiting dilutions such that each well contains at most cell; (3) depending upon the purpose of the study, optional screening for cells with antigen specificity; (4) cloning of V_H and V_L genes into vectors for sequencing and, when needed, expression of the respective antibody protein[45].

Single cell analysis has assisted in the discovery of novel therapeutic antibodies[49]–[52]. For example, single cell analysis has been used extensively to identify broadly neutralizing antibodies against rapidly evolving viruses such as HIV and flu[49], [53]–[56]. Unfortunately, single cell cloning has two major limitations. First, the throughput of analysis is limited to a few hundred cells per experiment. Second, single cell cloning is very time and materials intensive. To some extent, these limitations can be addressed by applying microfluidic and lithographic techniques which can increase cell throughput by three orders of magnitude[45], [48], [57]. However, such methods are costly and materials are not always readily available to all research institutions.

Next Generation Sequencing

Before the 21st century, DNA sequencing relied upon the method developed by Fred Sanger and then automated by Applied Biosystems and other companies[58]. Automated systems using Sanger sequencing could, at best, process only a few hundred sequences simultaneously[59]. In 2005, so called next generation sequencing (NGS) was introduced by 454 Life Sciences. This and other technologies for high throughput sequencing that soon followed enabled the parallel sequencing of 100s of thousands of DNA strands at a relatively low cost[60]–[62]. With NextGen sequencing, throughput has increased from 10^2 kbp/day in the late 20th century to more than 10^{12} kbp/day today[61]. Scientifically, next generation sequencing has facilitated the ability to perform high throughput genome sequencing, transcriptional analysis, epigenetic marker detections, and identification of genetic markers that correlate with disease[59].

Application of NGS methods provides a means for characterizing, at sufficient depth, a large fraction of sequences encoded by B cells comprising the immunoglobulin repertoire [8], [9], [11], [63]–[66]. In principle, NGS methods can theoretically yield sequence information from 10^6 B lymphocytes. In contrast to single cell cloning methods, sequencing B cell receptors by NextGen sequencing is rapid, requires relatively little material and cost, and is a powerful tool in both fundamental immunology research, and for therapeutic and antibody discovery purposes (Figure 1.5). The overall approach for deep sequencing immunoglobulin repertoires is as follows. First, a desired B cell subset is isolated by FACS from either blood or a particular tissue of interest. Second, cells are lysed and cDNA is synthesized from mRNA encoding B cell receptors. Third, V genes within the cDNA pool are PCR amplified and prepared for next generation sequencing. Finally, bioinformatics methods are required to analyze the sequenced repertoire. While most steps in this pipeline are standardized, the final task of bioinformatics analysis will

vary greatly for each experimental study. Common methods of analysis can require the use of algorithms which perform multiple sequence alignments, nucleotide and amino acid sequence clustering, and robust statistical analyses[67]–[70]

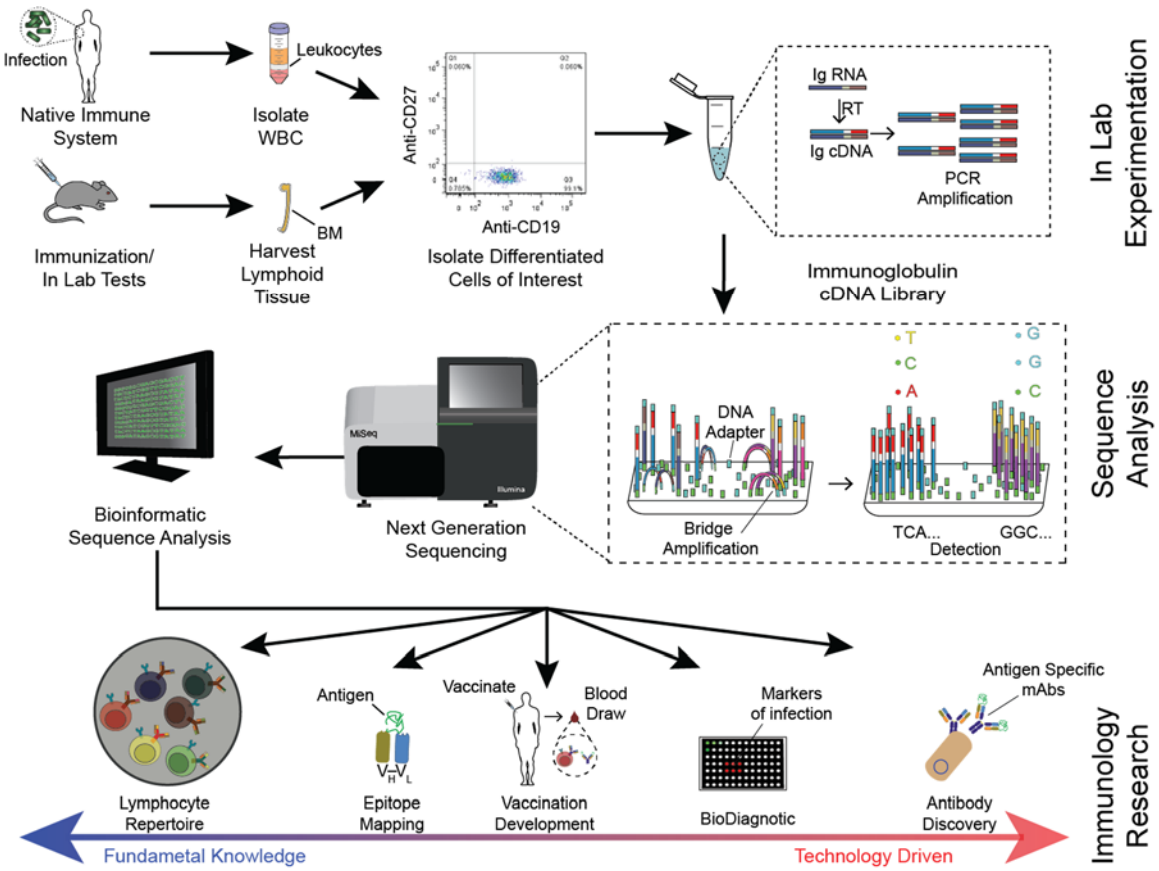


Figure 1.5: High throughput sequencing of immune repertoires

Deep sequencing the immunoglobulin expression profile from millions of B cells of different species has led to in depth quantifiable analysis of germline usage, CDR3 length, and amino acid composition[63], [71]–[75]. For instance, one of first applications of next generation sequencing in immunology was the determination of the

immunoglobulin repertoire in zebra fish[11]. Using next generation sequencing, they were able to characterize the patterns of VDJ usage in the naïve zebra fish repertoire and, moreover, identify immunoglobulin sequences that were present across multiple fish[11]. These investigations have expanded our current knowledge concerning the distribution and mechanisms associated with the immune repertoire from both healthy and unhealthy individuals[76]–[78]. Additionally, high-resolution analysis of repertoires may be applied in the identification of biomarkers indicative of viral infection such as Dengue or cancer malignancies[79], [80]. Finally, next generation sequencing has proven useful for the identification of antigen specific antibodies following immunization[9].

Since this shift in immunology towards genomic analyses using next generation sequencing, various sequencing platforms have emerged, each proclaiming benefits of improved sequencing and accuracy. The choice of NGS platform will define the technological limitations of an experiment, how the data must be processed before analysis, and the types of non-biological errors that will be introduced after sequencing. The following section discusses NGS technologies commonly used for sequencing immunological repertoires.

METHODS AND PLATFORMS FOR NEXT GENERATION SEQUENCING

While each individual platform varies with respect to number of sequences generated per run, read length, error rate, and total cost, all next generation sequencing platforms have three steps in common: (1) Library generation and amplification without bacterial cloning; (2) Conjugation of DNA library to a solid support such as glass slide or microbead; (3) Application of a method for parallel DNA synthesis and the identification of incorporated bases during synthesis[61], [62].

Library preparation

The first step in most next generation sequencing platforms, such as Roche 454 or Illumina MiSeq, is the addition of small adapter nucleotide sequences to the ends of the amplicons which act as primers for downstream polymerase and extension reactions. Depending on the technology, template libraries for sequencing can either be directly sequenced (single molecule detection) or clonally amplified. Clonal amplification methods are more readily used because they afford a lower error rate[61]. Two common techniques for clonal amplification are emulsion PCR and solid phase amplification (Figure 1.6)[61].

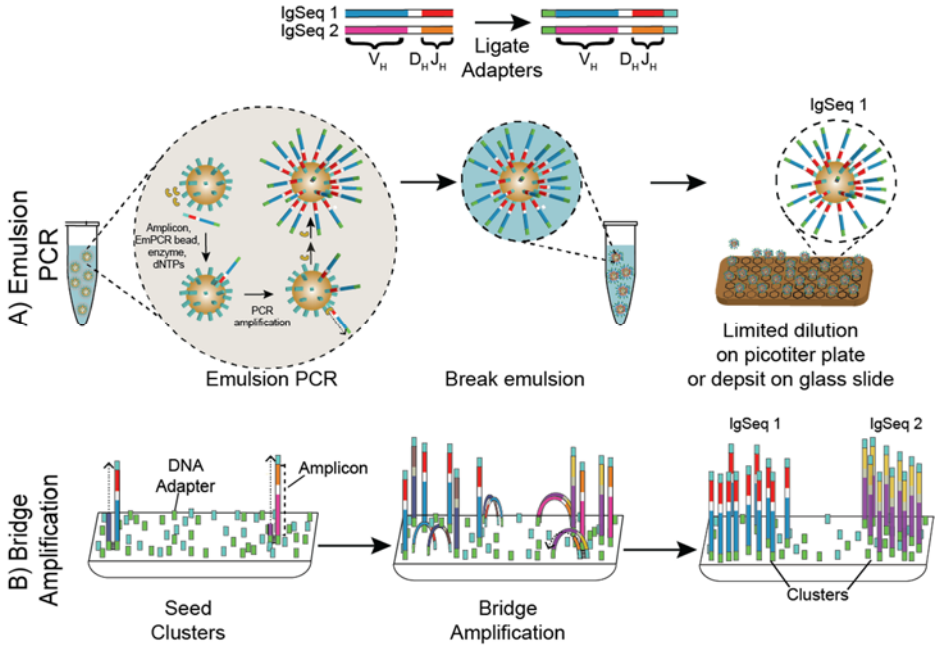


Figure 1.6: HTS methods for clonal amplification

A) Emulsion PCR: DNA is captured on beads and encapsulated in oil-water emulsion mixture containing all reagents (i.e. dNTP and polymerase) necessary for PCR amplification. PCR amplification is performed within droplets, generating beads containing thousands of copies of the original amplicon. Emulsion is broken and individual beads are deposited in picotiter plates or glass slides for downstream sequencing. B) Individual amplicons are deposited sparsely on solid support. Amplicons are used to prime clusters and immobilize template on surface. Clusters of DNA strands are created using bridge amplification of immobilized template with neighboring primers.

In emulsion PCR, individual template DNA molecules are ligated with adapter sequences and then placed in oil-water emulsion mixtures. Within the emulsion mixture, single stranded DNA molecules, via their ligated adapters, are captured onto beads under conditions that favor one DNA molecule per bead. PCR amplification is performed in individual emulsion droplets. After PCR amplification, EmPCR beads will contain thousands of identical copies the original captured DNA strand[81]. After amplification, the emulsion is broken and EmPCR beads containing PCR amplified sequences are immobilized either in a picotiter plate or on a solid glass surface for DNA sequencing and detection.

In contrast to emulsion PCR, solid phase amplification generates clonally amplified DNA clusters directly on the glass surface of the flow cell. Initially, this glass surface is coated with a high density of oligonucleotides that are complementary to the adapter sequences that were ligated to template DNA strands. The oligonucleotides will immobilize adapter-ligated DNA strands deposited distantly on the glass slide[61]. The captured DNA strand is subsequently used as a template for the elongation of its complementary strand via PCR. The high density of neighboring oligonucleotides, covalently bound to the glass surface serve as primers for further amplification of the DNA template and its complementary strand. Therefore, bridge amplification results in the generation of a dense clusters of amplicons seeded by the original template strand. In principle, because the templates are deposited in spatially distinct areas along the slide, there is no contamination by clusters of neighboring templates. Amplified "template clusters" are subsequently subjected to DNA sequencing.

Sequencing and Detection

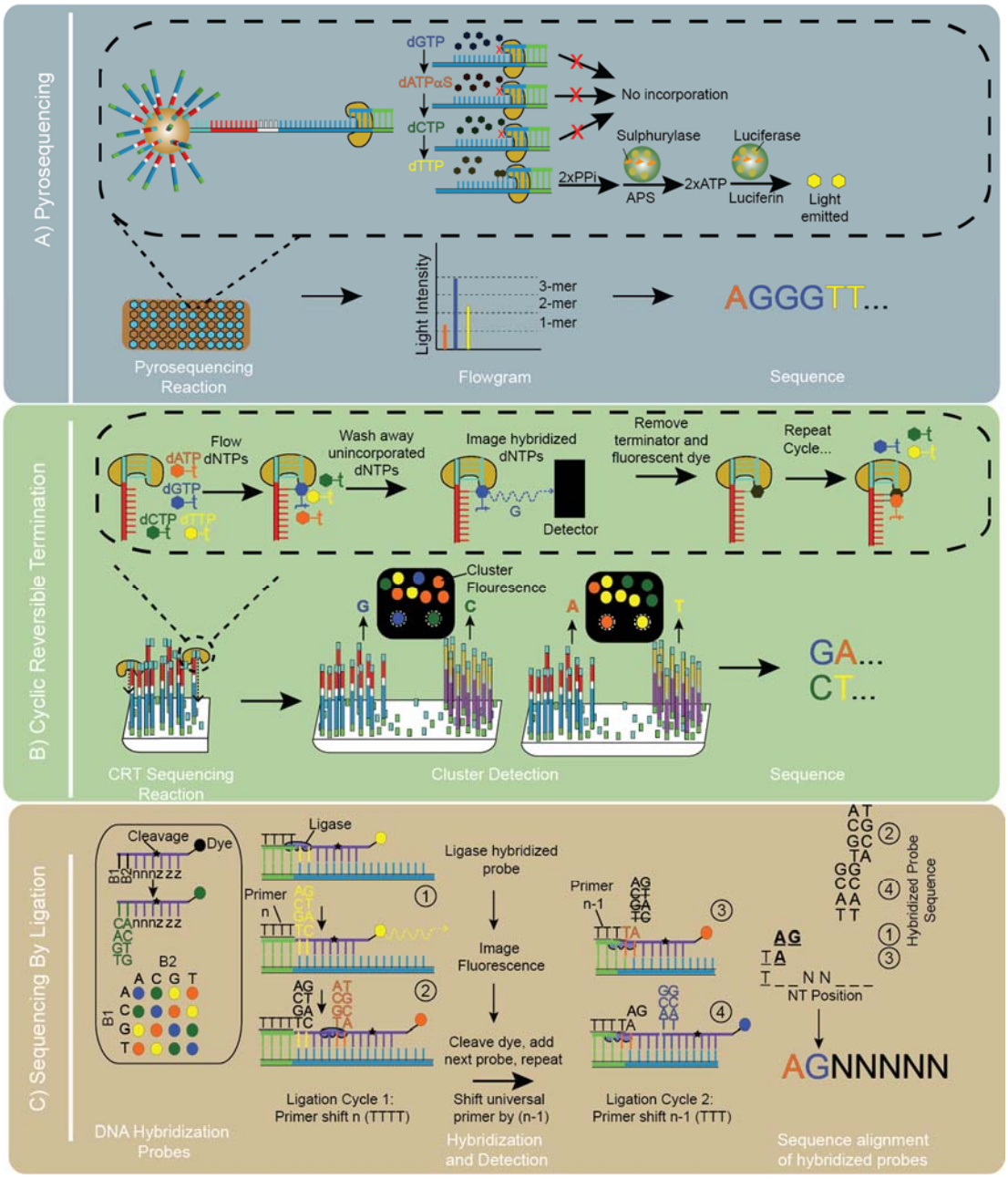


Figure 1.7: HTS methods for sequencing and imaging clonally amplified templates

A) Pyrosequencing: Indirect detection of nucleotide incorporation via pyrophosphate reaction with Sulphurylase and Luciferase. B) CRT: Direct detection of nucleotide incorporation via fluorescent dNTPs. C) Sequencing by ligation: Hybridization of fluorescent probes.

NGS platforms use a variety of techniques for DNA sequencing. These techniques include, but are not limited to the indirect detection of nucleotide incorporation, direct detection of nucleotide incorporation via fluorescent dye, and sequencing by ligation (Figure 1.7)[61]. Pyrosequencing is an example of a method where sequencing relies on indirect detection of nucleotide incorporation[61]. During pyrosequencing, each nucleotide is individually washed over the library of amplicons. Successful, incorporation of a complementary nucleotide into the growing DNA sequence results in the release of a pyrophosphate group. ATP sulfurylase and luciferase enzymes present in the solution detect the release of pyrophosphate and emit light whose intensity is directly proportional to the number of nucleotides incorporated during that cycle.

A common method of direct detection of nucleotide incorporation is called cyclic reversible termination (CRT)[61]. During CRT, DNA polymerase incorporates a fluorescent nucleotide that is complementary to the current base of the primed template. Each fluorescent nucleotide contains a reversible terminator group which prevents any other nucleotide from hybridizing to the template. Upon incorporation, a detector records which fluorescently labeled nucleotide hybridized to the template base. Next, the terminator and dye are removed from the incorporated base, and the cycle continues iteratively for the remaining bases.

Finally, sequencing by ligation does not utilize a polymerase for DNA sequencing. Instead, this method relies upon DNA ligase which inserts fluorescently labeled DNA probes that hybridize to complementary sequences adjacent to the primed template. Probes that do not hybridize to the sequence are subsequently washed away.

NGS Platform	Template Preparation	Sequencing Method	Sources of Sequencing Error	Read Length (Avg bp); Error rate (mutations per bp)	Mb/ day; Cost	Pros	Cons
454 (Roche)	Emulsion PCR; Beads sequenced in picotiter plates	Pyro-sequencing	Contamination of more than on amplicon on a single bead; PCR amplification error; inaccurate sequencing of homopolymers; Pyrosequencing interference caused by sequencing nearby amplicons	>400; 10^{-3}-10^{-4}	750; High	Long read length; Fast run times; Lowest average mismatch error rate	Low throughput; High rate of insertions/deletions; Reagents are expensive; Sequencing protocol is cumbersome
Solid	Emulsion PCR; Sequenced on glass slide	Sequencing By Ligation	PCR amplification error; dephasing; decline of sequencing accuracy with increased amplicon length	<200; 10^{-2}-10^{-3}	5000; Low	Low cost; Low homopolymer error rate	Long run times
MiSeq/HiSeq (Illumina)	Bridge amplification PCR	Reversible termination	PCR amplification error; Sequencing interference caused by sequencing nearby clusters; dephasing; incorrect base-calling	>2x250; 10^{-2}-10^{-3}	5000; Low	High processivity; Highly available	High average mismatch-error rate

Table 1.2: Comparison of currently available NGS sequencing platforms

Table 1.2 summarizes NextGen sequencing platforms that are currently available and lists their respective advantages and disadvantages. Since 2005, two platforms have dominated mainstream DNA sequencing: 454 pyrosequencing by Roche and Illumina HiSeq and MiSeq sequencing. 454 sequencers return longer sequence reads with an overall lower error rate of base-pair mismatches at the expense of higher sequencing costs, lower processivity, and a higher probability of errors when sequencing homopolymer strands of DNA. Illumina is currently the leading technology in next generation sequence because it can support more than 10^6 reads per plate at a low cost. However, regardless of the technology, there are still many challenges that must be overcome when applying NGS in immunoglobulin repertoire analysis.

CHALLENGES INTRODUCED IN NEXT GENERATION SEQUENCING OF ANTIBODY REPERTOIRES

Although highly promising, many challenges must be addressed when using next generation sequencing for the analysis of immunoglobulin repertoires. A common experimental aim in this field is the identification of immunoglobulin sequences that correspond to unique somatic variations of the BCR. However, non-biological variation in the sequence can also be introduced by next generation sequencers which have intrinsic error rates ranging from 10^{-2} to 10^{-4} mutations per base pair[82]. Furthermore, PCR amplification of the cDNA library prior to sequencing introduces base pair mismatch error, insertion and deletion errors, and can skew the cDNA transcriptome. The latter is caused by the inability to amplify individual transcripts which do not anneal preferentially to the mixture of degenerate primers[83]. Thus, distinguishing biologically relevant sequence variation from experimental error can be difficult.

Proper experimental design can address some of these challenges. Sequencing technical replicates of cDNA libraries amplified with two or more primer sets can be useful for determining the statistical significance of differences observed in downstream analyses of non-technical replicates[11]. Additionally, sequence error introduced during PCR amplification and high throughput sequencing can be minimized by annealing random oligonucleotide barcodes to cDNA transcripts before PCR amplification. These barcodes are used, post-sequencing, to align amplicons derived from the same template, and correct for any non-biological sequencing errors that were introduced[84]. Finally, spiking known immunoglobulin DNA sequences into samples can be used to establish the sensitivity and sequencing depth of a repertoire study[85].

Regardless of experimental design, bioinformatics analysis is essential for deconvoluting the humoral immune response. Researchers are heavily reliant upon analysis using openly available software packages or custom-made bioinformatics pipelines. This requirement poses a significant challenge for researchers who lack strong computational backgrounds, and yet have an overabundance of sequence information. Many algorithms for immunology research are not designed to handle the information currently generated by next generation sequencing. More importantly, the lack of standardization in this field makes it difficult to select the proper methods of bioinformatics analysis, and carry out meta-analyses of data compiled from various laboratories.

The following chapters presents tools which address such challenges in big data and repertoire analysis. First, in chapter 2, methods are presented for the statistical analysis and comparison of similarity between multiple immunoglobulin repertoires. Second, chapter 3 introduces a novel algorithm for rapidly aligning immunoglobulin sequences to their respective V_H genes. Finally, in chapter 4, a cloud-based immunoglobulin analysis pipeline has been designed to address the lack of a validation, transparency and

reproducibility of repertoire studies. In addition, this pipeline has been designed around a database for the deposition and sharing of computational resources across the immunological community.

Chapter 2: A systems analysis of immunoglobulin repertoires in the lymphoid tissues of immunized mice

INTRODUCTION

The ability of the adaptive immune system to produce antigen specific antibodies following antigen challenge is critical for the elimination of pathogenic cells, and is of great interest for antibody discovery. The humoral adaptive immune response is dependent upon the successful selection and maturation of antigen specific immune cells from a diverse population of B cells that originate from developing precursor pro-pre-B cells in the bone marrow [86], [87]. During differentiation in the bone marrow, pro-pre-B cells undergo diversification of their B cell receptors (BCRs) by mechanisms such as the somatic recombination of the variable (V), diversity (D), and joining (J) gene segments, and the introduction of junctional diversity between recombined gene segments[88].

Once differentiated, naïve B cells migrate to peripheral lymphoid tissues and survey for "non-self" molecules that are present in lymphatic fluid or presented on the cell surface of antigen-presenting cells. Antigen-activated B cells form germinal centers and undergo rounds of division and affinity maturation via somatic hypermutation of their B cell receptors. These clonally expanded antigen-experienced B cells differentiate into either memory B cells or terminally differentiated antibody secreting plasma cells (PC), which secrete copious amounts of antibodies in serum[88]–[97]. Thus, especially in the case of an immunized individual, the immune repertoire of B lymphocytes post affinity maturation can become skewed towards clonally related B cells, termed clonotypes, that presumably originated from a particular naïve B cell precursor[86], [87], [98]–[100].

B lymphocytes circulate extensively through blood and lymphoid tissues. However, the distribution of these circulating lymphocytes within individual tissue compartments is not well understood. For example, it has not been determined whether

populations of clonally related B lymphocytes reside preferentially within a single lymphoid tissue, or are distributed evenly across multiple tissues. Further, it is not known whether differentiated plasmablasts distribute within secondary lymphoid tissues or rapidly home to the bone marrow for long term survival. Therefore, a particularly interesting issue is whether B clones encoding identical or highly similar antibody sequences are distributed equally among lymphoid tissues.

These questions may be addressed by employing next generation sequencing to survey the antibody repertoires of individual lymphoid tissues. Rigorous statistical methods can be used to compare the "degree of polarization" (i.e. prevalence of highly expanded clones) within specific tissues, as well as the overlap among the immunoglobulin repertoires of lymphoid organs following immunization[101]–[104]. It should be noted, however, that the comparison of B cell repertoires in various organs will be influenced by the immunization protocol. For example, the immune system's response to antigen can be affected by the route of immunization, the nature of the antigen and adjuvants used during immunization, and the time following immunization when organs are harvested for analysis.

Here we investigated the antibody repertoires encoded by antibody secreting cells ($CD138^+$ plasmablasts and plasma cells) from the bone marrow, spleen, and lymph nodes of five different BALB/c mice immunized with the antigen hen-egg-lysozyme (HEL). We present methods for the comparative analysis of V_H gene expression in each population. Particular emphasis of the work presented here illustrates analytical approaches for discerning relevant features of a repertoire from large sets of sequence data. In addition we present data visualizations methods for conveying the biological relevance of the analysis. Finally, we report that in animals where the same antibody sequences were

present at a high frequency in all tissue compartments, the top most expanded V_H sequences encode for antigen-specific antibodies.

DEFINITION OF TERMS USED IN THIS STUDY

The following describes the meaning of terms used throughout this study:

a) *V_H gene sequence*: Refers to a unique full length immunoglobulin sequence.

b) *Sequence counts*: Refers to the number of times we observe a unique V_H gene sequence in our data.

c) *V_H Clonotype-cluster*: For this study, a clonotype-cluster refers to a group of V_H gene sequences that encode for the same V gene segment and CDRH3 amino acid sequence. Biologically these sequences are believed to be clonally related B cells that differentiated from the same parent naïve B cell.

d) *Unique clonotype*: The unique clonotype refers to the unique CDRH3 sequence used to cluster V_H gene sequences into clonotype-clusters. The number of unique clonotypes corresponds to the total number of clonotype-clusters.

e) *Clonotype-cluster size*: Refers to the total number of sequence counts that are grouped into the respective clonotype-cluster.

f) *Singleton*: Refers to clonotype-clusters or unique V_H gene sequences with a respective clonotype-cluster size or sequence count of one.

g) *Non-Singleton*: Refers to clonotype-clusters or unique V_H gene sequences with a respective clonotype-cluster size or sequence count greater than one.

h) *Tissue-specific clonotype*: Refers to a clonotype-cluster comprised of V_H sequences identified in only a single lymphoid tissue.

i) *Tissue-shared clonotype*: Refers to a clonotype-cluster comprised of V_H sequences from two or more lymphoid tissues.

j) *Tissue Polarization*: A tissue that is highly polarized would be indicative of a repertoire containing a small number of unique clonotype-clusters that are observed at a very high frequency; conversely, a non-polarized tissue or highly diverse population would be indicative of a repertoire consisting of numerous unique clonotypes present at very low frequencies.

MATERIALS AND METHODS

Experimental bench work

The wet lab experimental methods described in this section were performed by Kam Hon Hoi.

Immunization regime

Five female BALB/c mice were obtained at six weeks of age from The Jackson Laboratory (Bar Harbor, ME). The mice were housed in the Animal Resource Center for the University of Texas at Austin. All experimental procedures were conducted under the guidelines of the university's Institutional Animal Care and Use Committee (IACUC # AUP-2010-00089). Six-week old female BALB/c mice were primed with 30 μg of Hen-Egg-Lysozyme (HEL) (Sigma Cat# L6876-25G) and 25 μL of Complete Freund's Adjuvant (CFA) (Thermo Scientific Prod# 77140) subcutaneously. Twenty-one days post primary immunization, the mice were boosted with 25 μg of HEL and 25 μL of Incomplete Freund's Adjuvant (IFA) (Thermo Scientific Prod# 77145) intraperitoneally. Seven days after the boost, the mice were injected with a further 25 μg of HEL and 25 μL of IFA intraperitoneally.

Immune cell isolation and plasma cells enrichment

Seven days after the second immunization boost, the mice were euthanized with carbon dioxide asphyxiation. Spleen, femur and tibia bone marrow, and a total of 6-8 lymph nodes (2 brachial, 2-4 lumbar, and 2 mesenteric) were collected for immune cell isolation. Single cell suspensions of secondary lymphoid organs were prepared in ice-cold buffer no.1 (sterile-filtered PBS with 0.1% BSA and 2 mM EDTA) separately for each lymphoid organs from each animal. A detailed description of bone marrow isolation was shown in a previous report[101]. Isolation of single cells from the spleen and lymph nodes were performed as follows. Tissues were mechanically teased apart. To separate single cells from connective tissue, teased samples were ground on a 70 μM mesh filter. PBS was used to resuspend single cells and subsequently filter ground cells through the mesh filter. Plasma cells and plasmablasts are the only B cell subsets that express CD138. Therefore, to isolate plasma cells and plasma blasts from other B cells, $CD138^+$ cells were enriched using anti-mouse CD138-biotin (BD Pharmingen Cat# 553713) and Dynabeads M-280 streptavidin (Invitrogen Cat# 112.05D).

RNA extraction and cDNA generation

$CD138^+$ enriched cells from each lymphoid tissue were lysed separately using the TRI Reagent (Ambion Cat# AM9738). Total RNA was isolated according to the manufacturer's RiboPure Kit protocol (Ambion Cat# AM1924). First strand cDNA of IgG specific immunoglobulin sequences were generated from a minimum of 100 ng of isolated total RNA using SuperScript RT II kit (Invitrogen Cat# 18064-022) and primers that bind specifically to the CH1 region of the IgG constant chain. PCR amplification of the first strand cDNA was performed using conditions described previously[101] and a validated V_H gene primer mix [105]. PCR products were gel-purified and subsequently submitted for 454 next generation sequencing.

ELISA assays of serum titer

For at least 16 hours at 4 ^{o}C, Polystyrene 96 well plates (Costar #3590) were coated with 100 µl of Hel at a concentration of 8 µg/mL. To separate plasma cells from blood, 20 µL of blood from the terminal bleed of the mice was centrifuged on a bench-top centrifuge at 5000 RPM for five minutes. Anti-HEL serum antibodies in the supernatant were determined by ELISA. Goat anti-mouse IgG HRP (Invitrogen Cat# 62-6520) conjugated antibody was used as the secondary antibody and TMB substrate (Thermo Scientific Prod# 34028) was used to develop the plate. Serum titers were determined by identifying the dilution level where absorbance at 450nm diminished to background levels.

Antibody expression

The V_H gene amino acid sequences corresponding to the top five most highly abundant clonotypes in the highest titer mouse (Mouse 23), were used to construct synthetic genes (based on E. *coli* optimized codon sequences). The five synthetic V_H genes (Table B.5) were cloned into individual pFAB-S plasmids. Cloned V_H genes were combinatorially paired with the cDNA library of V_L light chains isolated from $CD138^+$ cells in the mouse 23 bone marrow. The combinatorial V_H-V_L paired library was transformed into Jude-1 *E. coli* cells. Subsequently, 96 colonies were individually picked and grown in 96 well plates and screened by ELISA. ELISA assays were performed by coating polystyrene plates (Costar #3590) overnight at 4 ^{o}C with 8 µg/mL of HEL or 8 µg/mL of BSA (as an unrelated antigen control). After initial screening of potential antigen-specific binders, competitive ELISAs were performed to confirm antigen-specificity of the heavy/light chain pairings. For competitive ELISA, antigen-specific binders were preincubated with 80 µg/mL of soluble HEL for one hour. After washing away soluble HEL and other molecules in plate solution, the amount of bound antibody

remaining in the well was measured using ELISA as described above. Samples with loss-in-signal after competitive ELISA suggested heavy-light pairs with strong affinity for HEL.

In-silica sequence analysis

Raw data processing

In total, 15 cDNA samples were submitted for 454 pyrosequencing (Table 2.2). We assumed sequences less than 350 base pairs were not full length heavy chain sequences containing a V_H, D_H, and J_H gene segment. Therefore, sequences less than 350 base pairs were removed from the analysis. Sequences that were longer than 350 base pairs and also passed internal Roche 454 quality filters were analyzed using IMGT High V-Quest immunoglobulin analysis software[67]. First, full length DNA sequences containing stop codons or lacking an identifiable V_H, D_H, and J_H gene were considered non-productive sequences and filtered out. Full length immunoglobulin sequences were analyzed for the following features using results extracted from the IMGT High V-Quest output files: V_H gene family, CDRH1, CDRH2 and CDRH3 amino acid sequences, full length nucleotide sequences, and the degree of somatic hypermutation as compared to the sequence's predicted germline assignment.

V_H Gene family usage

In mice, the V_H germline genes are grouped by sequence similarity into 16 gene families. For each lymphoid tissue, we calculated the distribution of V_H gene family usage across sequences. Chi-square analysis was used to calculate the similarity of V_H gene family usage across multiple lymphoid tissues. For each mouse, V_H families observed at very low frequencies (sequence count < 5) were binned together before comparing usage between lymphoid tissues. Pairwise tissues with a reduced χ^2 value < 3.8 (p-value <0.05) were assumed to demonstrate an identical distribution of V_H gene usage.

Clonotype correlation analysis

Pearson correlation analysis was used to compare the frequency of clonotype-clusters shared between pairwise tissues. For each pairwise tissue comparison, a Pearson correlation coefficient was calculated from clonotype-clusters present in both lymphoid tissues. Finally, using Fisher r-to-z transformations, we calculated the statistical significance of correlation coefficients between different pairwise tissue comparisons.

Clonotype diversity analysis

Diversity Index (x)	Function (x=)	True Diversity in terms of Diversity Index ($D_T(x)$)	True Diversity in terms of frequency ($D_T(f)$)
General function (q = order)	$\frac{(1-\sum_{i=1}^{S} f_i^q)}{(q-1)}$	$\left[\left(1-\ (q-1)x\right)\right]^{1/(1-q)}$	$(\sum_{i=1}^{S} f_i^q)^{1/(1-q)}$
Species Richness (analogous to q=0)	$\sum_{i=1}^{S} f_i^0$	x	$\sum_{i=1}^{S} f_i^0$
Shannon entropy (q=1)	$-\sum_{i=1}^{S} f_i \ \ln(f_i\ )$	Exp(x)	$\text{Exp}[-\sum_{i=1}^{S} f_i \ \ln(f_i\ )]$
Gini-Simpson index (q=2)	$1-\sum_{i=1}^{S} f_i^2$	$\frac{1}{1-x}$	$\frac{1}{\sum_{i=1}^{S} f_i^2}$

Table 2.1: Common indices of diversity and normalized true diversities

Diversity index quantifies population diversity. Sensitivity of diversity to frequency, f_i, is represented by the order of diversity, q. True diversity is the normalized representation of the diversity index. Table adapted from Jost, 2006.

We wanted to quantify clonotype-cluster diversity or, conversely, polarization in each tissue. Therefore, we referred to methods used in ecology for characterizing the

biological diversity of populations. Table 2.1 provides a list of diversity indices (described in detail in Appendix B) commonly used to characterize diversity.

Diversity indices are monotonic functions in which a larger diversity index indicates a more diverse population. However, as explained in Appendix B, one caveat in using diversity indices is that, while they are monotonically increasing, they are not all linear. This non-linearity could be misleading when comparing populations because a population whose Shannon entropy diversity index is 10 is not twice as diverse as a population whose index is 5. Instead, it is useful to first normalize or linearize the diversity index into a term referred to as "true diversity" [106] (Table 2.1). The units of true diversity represent the "effective number of species" in a theoretical population consisting of an even distribution of species present at equal frequencies, and whose diversity index is equivalent to the diversity index in the non-normalized population (Appendix B). Most importantly, using this definition of effective number of species, a population whose true diversity is 10 can be considered to be twice as diverse as a population with a true diversity of 5. Equation 2.1 describes a generalized form of true diversity, D_T, as a function of both species frequency, f_i, and the diversity order, q[106].

$$D_T(q) = \left(\sum_{i=1}^{S} f_i^q\right)^{1/(1-q)} \quad \text{[Eq 2.1]}$$

Where:

D_T = True diversity/"Effective number of species"

q = The diversity order

S = The total number of unique species

f_i = The frequency of the current species *i*

In Equation 2.1, diversity is defined by the sum of species frequencies raised to the power of q. Hence, for each diversity index, the order of diversity, q, defines the sensitivity

of diversity to species frequency. Obtaining diversity measurements that do not bias diversity disproportionately by species frequency requires calculating true diversity using the first order of diversity (q=1). In this case, the diversity contributed by both rare and highly common species will be weighted directly by their frequency (i.e. each frequency is raised to a power of one)[106]. Equation 2.1 is undefined for q = 1, but its limit at q =1 is equivalent to the exponential of the Shannon entropy diversity index (Table 2.1 row 3).

For the purpose of this study, the diversity of each lymphoid tissue is defined by the number unique clonotype-clusters and their corresponding frequency in each tissue. Precisely, true diversity is calculated using Equation 2.1 where: 1) S is the number unique clonotype-clusters, 2) f_i is clonotype-cluster frequency in a specific tissue, and 3) q = 0 (which uses a species richness diversity index) or q = 1 (which uses a Shannon entropy diversity index). We used species richness q = 0 to demonstrate the number of unique clonotypes observed in each tissue; Shannon entropy was used to quantify diversity by accounting for clonotype-cluster frequency.

RESULTS

Experimental pipeline

We sought to investigate, in a quantitative manner, the V_H repertoires in the bone marrow (BM), spleen (SP), and lymph nodes (LN) of mice that demonstrated a varied degree of immune response against a model protein antigen. Figure 2.1 illustrates the experimental pipeline for comparing the V_H repertoire between different lymphoid tissues. Briefly, mice were immunized with the antigen Hen-egg-lysozyme (HEL), and 21 days later, given two subsequent rounds of booster immunization one week apart. Because of the high cost of antibody sequencing and the relatively large number of samples that were required for this study, a total of five mice were selected for analysis in this study.

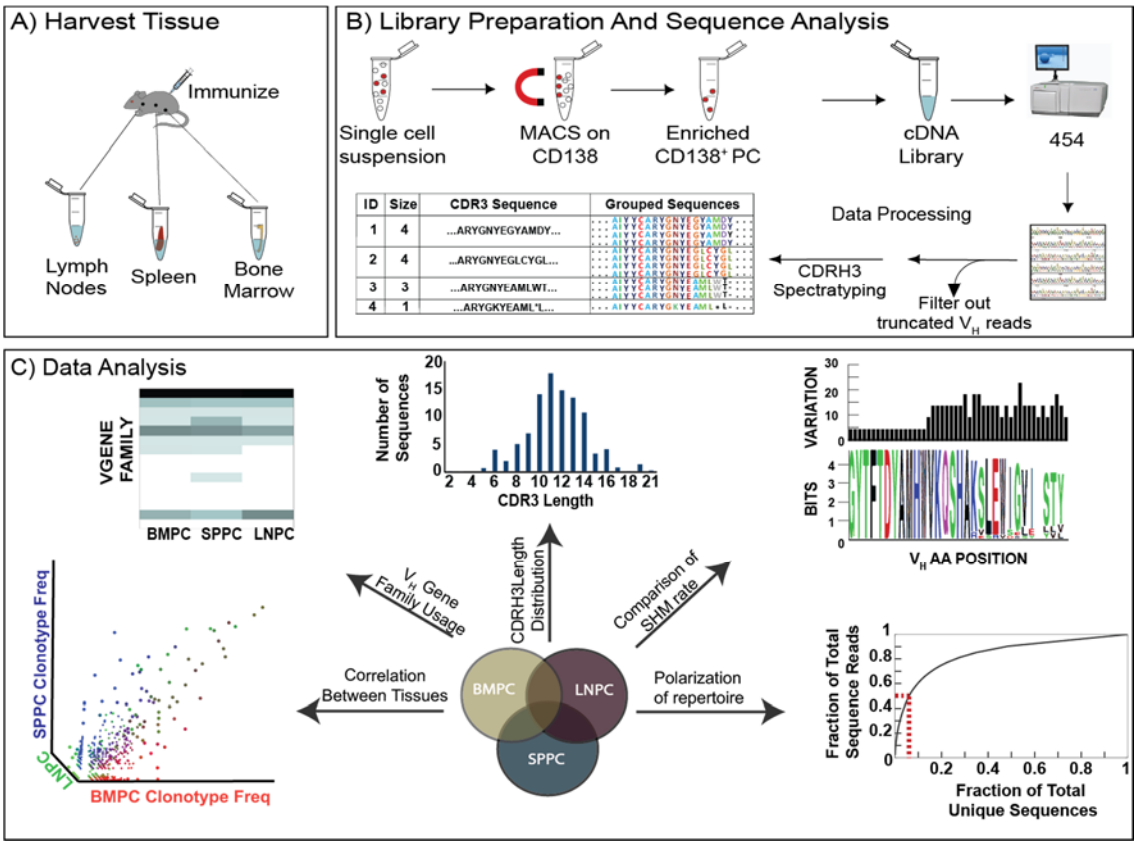


Figure 2.1: Schematic of the experimental approach used in analysis.

A) Tissues were harvested seven days post-secondary booster immunization. B) For each tissue, $CD138^{+}$ plasma cells were isolated using MACS with anti-mouse CD138-biotin. RNA was extracted, cDNA was prepared, and V_H genes were amplified and submitted for NextGen 454 sequencing. Quality processed raw reads were grouped into clonotype-clusters, defined as full length V_H genes that express the same V gene segment and CDRH3 amino acid sequence. C) Bioinformatics analyses characterized the distribution and overlap of the tissue repertoires.

ELISA analysis of mouse serum titers to the antigen HEL revealed that the immune response varied greatly between mice (Table B.3). Specifically, mouse 23, with a serum titer of (1/625000), had the strongest response against HEL, mouse 13 showed an intermediate response titer (1/125000), and mice 5, 8, and 9 showed a weak titer (1/25,000)

(Table 2.2). Mice were sacrificed seven days post-secondary booster immunization and $CD138^+$ cells were isolated from each lymphoid tissue using magnetic sorting. Sorted CD 138^+ cells were lysed and mRNA was isolated to synthesize cDNA and subsequently amplify IgG specific V_H genes using well validated primer sets that anneal upstream to the 5' end of the V_H gene and a 3' primer that anneals to the CH1 region of IgG (Tables B.1 and B.2).

	Tissue	# Sequence Reads	# Productive V_H Genes	# Clonotype Clusters	# Non-Singleton Clonotype Clusters
Mouse 5 (1/25000)	BM	20770	8659	1080	617
	LN	21668	9566	834	416
	SP	13995	4284	794	436
Mouse 8 (1/25000)	BM	22569	10113	1066	606
	LN	25588	9645	1128	578
	SP	25865	10784	1272	754
Mouse 9 (1/25000)	BM	16177	6374	833	477
	LN	20343	9199	707	369
	SP	14463	5271	830	429
Mouse 13 (1/125000)	BM	24876	14361	1579	1006
	LN	17609	7220	539	269
	SP	17191	9217	1170	746
Mouse 23 (1/625000)	BM	9767	4274	766	366
	LN	18287	7383	605	220
	SP	9452	3405	742	337

Table 2.2: Sequence reads returned by 454 sequencing

15 samples were submitted for 454 NGS. Full length V_H sequence were grouped into clonotype-clusters. Non-singleton clonotype-clusters contain >1 V_H sequence. Mouse 23 showed highest titer, shown in parentheses, against HEL.

In total, 15 V_H gene libraries were sequenced using 454 pyrosequencing and $2.2x10^5$ sequence reads were obtained. After removing truncated V_H genes, approximately 1.3×10^5 sequence reads contained productive V_H genes without stop codons and a CDRH3 sequence greater than three amino acids (Table 2.2). Finally, full length V_H genes that encode for the same V gene segment and CDRH3 amino acid sequence were grouped into clonotype-clusters.

Similarities in the V_H Repertoires of Mice

We first investigated the relationship between serum titer analysis for antigen-specific antibodies and potential bias in underlying biological mechanisms that encode the B cell receptor repertoire for antibody secreting $CD138^+$ B cells. Specifically, the distribution of germline V_H family genes, average CDRH3 length, and the degree of somatic hypermutation (SHM) were evaluated to help discern how V(D)J recombination, N/P addition, and SHM directed affinity maturation shape the repertoire. Average variation in these V_H repertoire features, such as genes containing very low or high amounts of hypermutation could indicate factors contributing to a weak or strong immune response, respectively.

As a first analysis of the data, we pooled together all V_H gene sequences encoded by $CD138^+$ antibody secreting plasma cells from the spleen, bone marrow, and lymph nodes of an individual mouse. On average, the statistical distributions of these biologically relevant parameters in the repertoires of the immunized mice were similar to previously reported observations of a mouse repertoire. More importantly, the parameters did not appear to have any relationship with serum titer. The distribution of CDRH3 length for all V_H sequences within an individual mouse was found to have a mean length of 11.61 amino acids and a standard deviation of 2.65 amino acids (Figure 2.2 A). SHM was calculated as

the number of amino acid differences between the V gene segment and its respective V_H germline sequence. The degree of SHM in the mouse ranged from 4-8 amino acid substitutions or between 1-3% of the V gene segment (Figure 2.2B). A non-parametric Kruskal-Wallis one-way analysis of variance statistical test, which determines whether at least one sample within a group is statistically different, indicated that there was no significant difference among all mice animals with respect to CDRH3 length and degree of somatic hypermutation. A non-parametric test was preferred for this analysis because, unlike ANOVA, it does not require the samples follow a normal distribution.

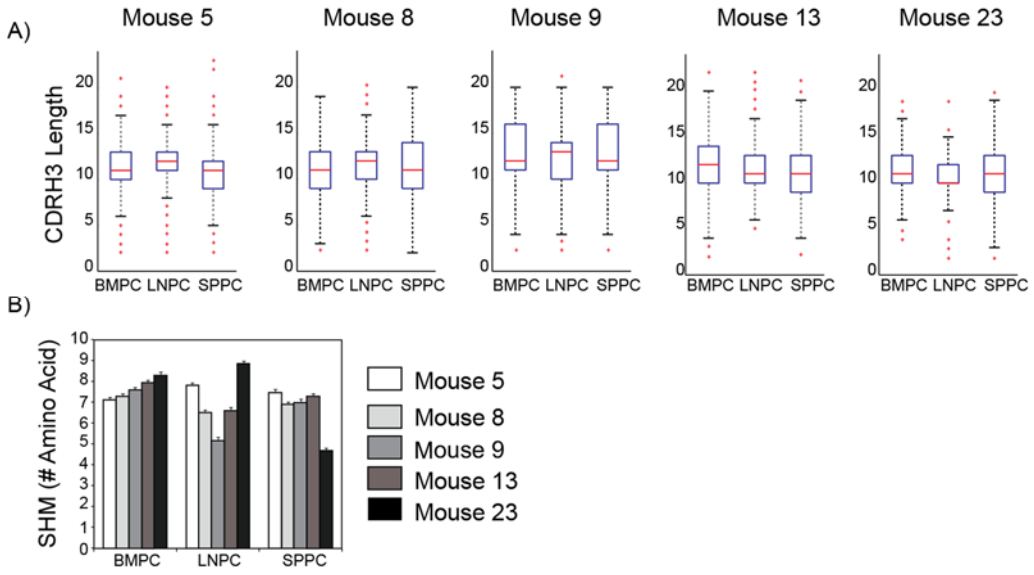


Figure 2.2: Comparison of CDRH3 length distribution and SHM across all mice

A) Box and whisker plot of CDRH3 length in each tissue. B) Average number of mutations in V_H genes as compared to its respective V_H germline gene.

The germline V_H gene segments of V_H sequences encoded by the $CD138^+$ repertoire were determined based on IMGT[67] assignment. We looked at the frequency of V_H sequences assigned to each of the possible 15 mouse families of V_H gene segments. The

rank-order (Spearman) correlation of V_H family usage was strongly correlated (>0.9) across all mice. The most commonly used V_H families were V_H1, V_H14, V_H5, and V_H2, which collectively contributed to at least 70% of the total sequences identified in each tissue sample set (Figure 2.3). V_H families found at a frequency < 0.5% in every mouse included V_H families 9, 11, 12, 13, and 15. No V_H 16 families were found in any mouse.

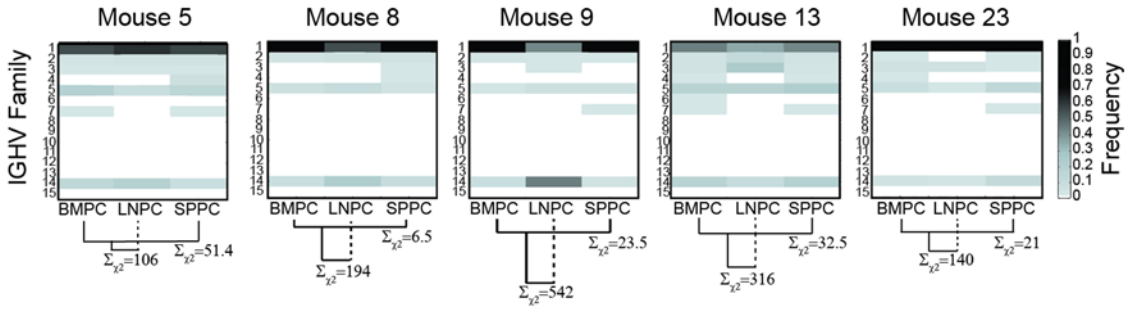


Figure 2.3: Comparison of V_H gene family distribution

Heat map of germline V_H gene family usage in each mouse tissue. Line brackets are proportional to square root of chi-square ($\Sigma\chi^2$) distance between pairwise tissues.

Comparison of the V_H repertoires within different tissues

Interestingly, pairwise comparison of the distributions of V_H gene family usage between the three lymphoid organs revealed potential biases in gene usage. One consistent trend observed in Figure 2.3 is that the bone marrow (BMPC) and spleen (SPPC) repertoires showed the smallest differences in V_H family usage in all mice. Specifically, chi-square analysis between the frequency of BMPC V_H family usage and SPPC V_H family usage indicated an average reduced $\Sigma\chi^2$ value of 27.7 +/- 16.3; in comparison, the average reduced $\Sigma\chi^2$ between the bone marrow and lymph node repertoires of all mice was 178.1 +/- 113.5 (Table 2.3). A Mann-Whitney rank test confirmed that this observation was statistically significant (p-value <0.01) and therefore the V_H family usage between the bone marrow and spleen were more similar as compared to the lymph node repertoire. The

spleen is a highly vascularized organ and B cells within this tissue are likely to migrate and equilibrate within other compartments such as the bone marrow. However, the distinct germline V_H gene frequency found among lymph node $CD138^+$ cells indicates that a population of differentiated B cells may remain resident within the lymph node, and not circulate to the bone marrow or spleen.

	Pairwise Tissue Comparison ($\Sigma\chi^2$)		
	BMPC vs. SPPC	**BMPC vs. LNPC**	**LNPC vs. SPPC**
Mouse 5	51.4	85.3	116
Mouse 8	6.5	142.3	132.1
Mouse 9	23.5	354.5	332
Mouse 13	32.5	222.5	214.8
Mouse 23	21	85.7	162.4

Table 2.3: Chi-square analysis of V_H gene family usage between tissues

This table lists the reduced sum of chi-square error when comparing the frequency of V_H gene families between pairwise lymphoid tissues.

Post immunization, antigen-activated B cells enter germinal centers within secondary lymphoid tissues and undergo clonal amplification and affinity maturation. Therefore, it is not surprising to find lymphoid tissue repertoires of immunized mice dominated by only a few clonally related V_H genes represented at very high frequencies. Initial analysis of the observed number of unique V_H sequences (sequence counts) in our datasets identified only a few non-singleton (sequence count > 1) V_H sequences (Figure B.1 A). Specifically, the distribution of unique V_H genes in each mouse had a median value of one count per V_H sequence (singleton reads); we did not identify an identical V_H sequence more than 11 times in any mouse. This infrequency of non-singleton sequences was most likely due to random sequencing errors introduced during PCR amplification and

high throughput sequencing. We clustered together sequences that encoded for the same germline V_H gene segment and CDRH3 amino acid sequence. We assumed that variation in the nucleotide sequence of clustered V_H sequences was either due to experimental sequence error or somatic variation among the antibodies expressed by clonally related B cells. Therefore, these clusters, which may represent groups of clonally related B cells having originated from the same parent naïve B cell, are referred to in this study as clonotype-clusters. In contrast to unique V_H sequences, we were able to identify numerous non-singleton clonotype-clusters in all mice. We next investigated how these clonotype-clusters were distributed throughout the bone marrow, lymph node, and spleen tissues.

The degree of polarization towards clonally expanded clonotypes was quantified in each of the lymphoid tissues. We expected that, post-immunization, biological processes such as affinity maturation would result in significant polarization of V_H gene repertoires towards a few clonally expanded clonotypes that show specificity for the antigen. In this study, polarization was defined by the diversity of V_H clonotype-clusters in each tissue. A highly polarized population would be representative of a V_H repertoire dominated by only a small population of clonotype-clusters and, thus, would not be considered very diverse.

Clonotype cluster diversity was calculated using "true diversity" defined above by Equation 2.1 (materials and methods). As described in the materials and methods, the diversity of a population can be quantified using a number of indices for diversity (Table 2.2). The equation for true diversity can transform each of these diversity indices into a function whose diversity index scales linearly with diversity (materials and methods). In Equation 2.1, true diversity is a function of both the species frequency, f_i, and the order of diversity, q. In this study, clonotype-cluster frequency (the ratio of the number of V_H sequences from a specific tissue grouped into a cluster to the total number of V_H sequences in the tissue) will represent species frequency, f_i. Moreover, as discussed in the methods,

the order of diversity, q, characterizes the sensitivity of true diversity to species frequency, or in this instance, clonotype-cluster frequency.

For example, we first calculated diversity using a zero order measurement for true

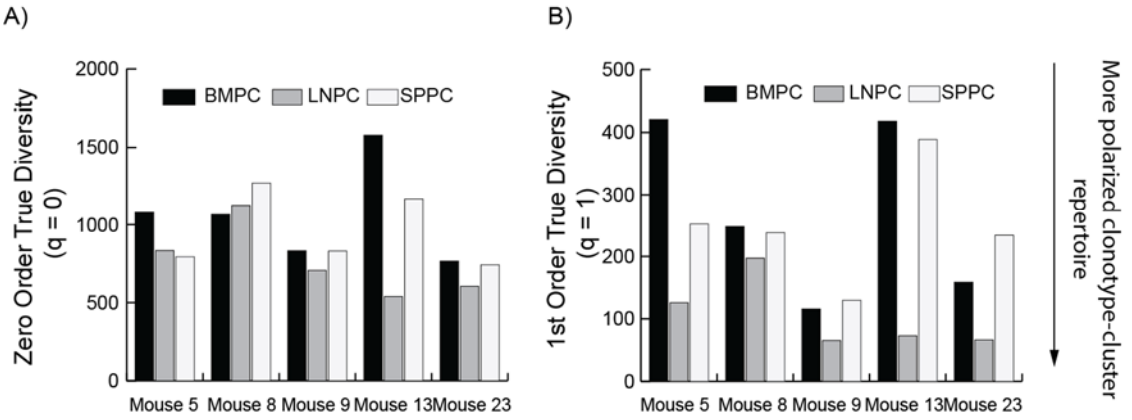


Figure 2.4: Diversity of each mouse lymphoid tissue with respect to V_H clonotypes

A) Zero order true diversity B) First order true diversity. This is also known as the exponential of Shannon entropy.

diversity (q = 0). In this instance, diversity is simply the number of clonotype clusters we identified in each tissue, and is insensitive to clonotype cluster frequency. For the set of the five animals studied here, the zero order true diversity, was approximately 1065 +/- 320, 762 +/- 232, and 961 +/- 241 respectively for the bone marrow, lymph nodes, and spleen (Figure 2.4A). A Mann Whitney rank test revealed that zero-order true diversity did not differ in a statistically significant manner among the three lymphoid tissues (p-value >0.05). This analysis indicates that the number of clonotype-clusters did not vary significantly between lymphoid tissues. This measurement of diversity, however, weights both small and very large clonotype-clusters equally. Consequently, small clonotype-

clusters will contribute to the measurement of diversity disproportionately to their respective frequency in the tissue compartments.

First order true diversity (q=1), on the other hand, is considered a more informative diversity measurement because it is not biased by highly promiscuous or rare clonotype-clusters (Equation 2.2).

$$\mathrm{Exp}[-\sum_{i=1}^{S} f_i \ \ln(f_i)] \text{ [Eq 2.2]}$$

Where:

S = Number clonotype clusters

f_i = clonotype cluster frequency

Equation 2.2 is equivalent to the exponential of the Shannon entropy diversity index (Materials and Methods, Table 2.1). Taking into account clonotype-cluster frequency in each tissue compartment and using first order true diversity (Equation 2.2) showed that, on average, the lymph nodes were half as diverse as the bone marrow or spleen (Figure 2.4B). This observation is supported by a Mann-Whitney rank test showing that the differences between true clonotype-cluster diversity in the lymph node to clonotype-cluster diversity in both the bone marrow and spleen are significant (p value=0.05 and 0.02, respectively) (Table B.4). There was no difference in clonotype-cluster diversity between the bone marrow and spleen tissues (*p*-value ~1). Based on this analysis, the lymph node repertoire is more polarized than both the spleen and bone marrow repertoires. More importantly, the majority of V_H gene sequences residing within the lymph nodes, which demonstrate low clonotype-cluster diversity in all mice, are probably derived from only a small number of clonotype-clusters.

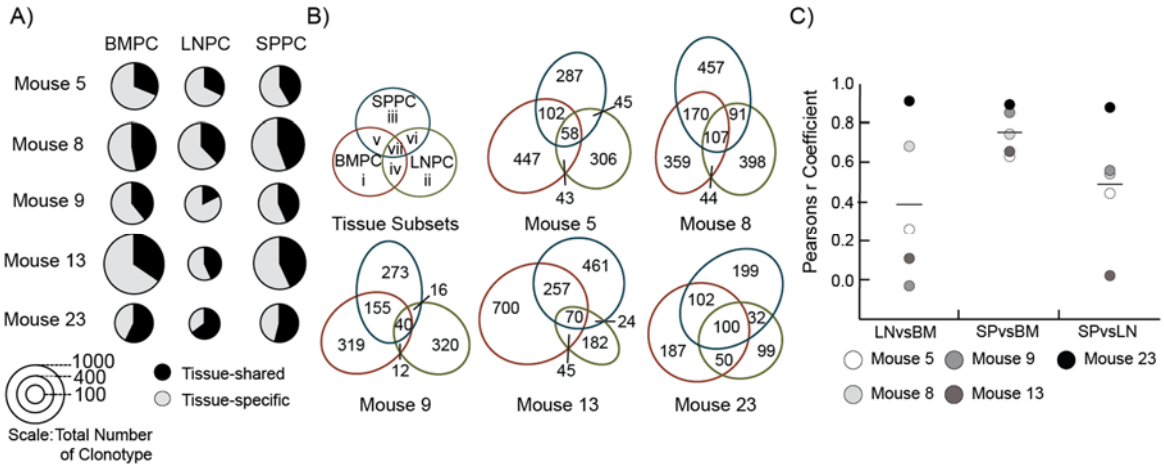


Figure 2.5: Distribution of non-singleton clonotype-clusters across lymphoid tissues

A) The area of each circle represents the total number of non-singleton clonotype- clusters that comprise a specific tissue compartment. The fraction of clonotype-clusters found only in that respective tissue (tissue specific) are colored gray. B) Area proportional Venn diagram of the overlap of clonotype-clusters shared between tissues. The area of each of the seven subsets is proportional to the number of clonotype-clusters within a subset. C) Scatter plot representing Pearson correlation coefficients of clonotype frequency between pairwise lymphoid tissues. Horizontal lines represents the mean pairwise correlation across all mice.

Clonotype-clusters observed within multiple tissues ("tissue-shared clonotypes") may represent clonally related B cells that circulated between multiple lymphoid tissues. On average, 25% of all non-singleton clonotype-clusters comprised V_H genes identified in multiple tissues ("tissue-shared clusters") (Figure 2.5 A). Following previous analyses, the repertoire of clonotype-clusters in the lymph nodes contained the least overlap with both the bone marrow and spleen repertoires. The majority of "tissue-shared" clonotype-clusters were comprised of V_H genes present in all three tissues, or V_H genes found only in the bone marrow and spleen repertoires, but not the lymph nodes (Figure 2.5B).

Consequently, it was not surprising to find that the expression of V_H genes comprising "tissue-shared" clonotype clusters were, on average, more correlated in the bone marrow and spleen repertoires. Specifically, when comparing spleen to bone marrow (SPvsBM), lymph node to bone marrow (LNvsBM), and spleen to lymph node (SPvsLN), the average Pearson correlation coefficients of clonotype frequency was 0.75 +/- 0.12, 0.38 +/- 0.39, and 0.48 +/- 0.31 respectively (Figure 2.5 C).

	Relative rank of Pearson correlation coefficients		
Mouse	**BMPC vs. SPPC**	**BMPC vs. LNPC**	**LNPC vs. SPPC**
Mouse 5	1	2.5	2.5
Mouse 8	1.5	1.5	3
Mouse 9	1	3	2
Mouse 13	1	2.5	2.5
Mouse 23	1	1	1

Table 2.4: Ranking of Pearson correlation coefficients from pairwise comparisons

We used the Fisher R-to-Z transformation test to determine whether correlation coefficients between pairwise comparisons were statistically different. The table shown summarizes the ranking of correlation coefficients based on the results of this analysis. A ranking of 1 represents the tissue pair with the best correlation. If the p-value of the r-to-z test was > 0.01 then the rankings for the two groups of pairwise tissues were averaged (i.e. 1.5 or 2.5).

This trend, in which the V_H repertoires from the bone marrow are spleen appear more correlated, is statistically significant across all mice. For each analysis, we used the Fisher R-to-Z-transformation to determine whether the Pearson correlation coefficients observed between the bone marrow and spleen V_H repertoires could be considered statistically greater than comparison with the lymph node V_H repertoire. For example, with respect to mouse 5, Pearson correlation between SPvsBM was 0.63 whereas the Pearson

correlations for LNvsBM and SPvsLN were 0.26 and 0.44, respectively (Figure 2.5 C). The Fisher R-to-Z transformation shows that SPvsBM correlation (r=0.63) is significantly higher (p-value ~ 0.007) than the correlation between the spleen and lymph nodes (r=0.44). On the other hand, the correlation between the spleen and lymph nodes (r=0.44) is not significantly higher (p-value ~ 0.08) than the correlation between the bone marrow and lymph nodes LNvsBM (r =0.26). Fisher transformation analysis revealed that, with exception to mouse 23 and mouse 8, the V_H repertoires of the bone marrow and spleen (SPvsBM) were consistently the most correlated lymphoid tissues (Table 2.4).

Highly abundant V_H clonotype-clusters, represented by a large cluster size, might reflect antibodies elicited in response to recent immunization with the antigen HEL. Therefore, we investigated further the distribution of the 30 most prevalent clonotype-clusters in the bone marrow, lymph nodes, or spleen. In mice 5, 8, 9, and 13 we found that approximately 28/66 (42%), 29/65 (44%), 21/69 (30%), and 22/71 (31%) clonotype clusters are comprised of V_H sequences from all lymphoid tissue datasets.

Notably, in mouse 23, which displayed the highest serum titer for HEL antigen, 36/48 (75%) of the most abundant clonotype clusters are comprised of V_H sequences from all lymphoid tissues datasets. The frequency of clonotype-clusters varied between lymphoid compartments. Figure 2.6 succinctly summarizes both the overlap and respective frequency of highly prevalent V_H clonotype clusters within the bone marrow, lymph node, and spleen repertoires.

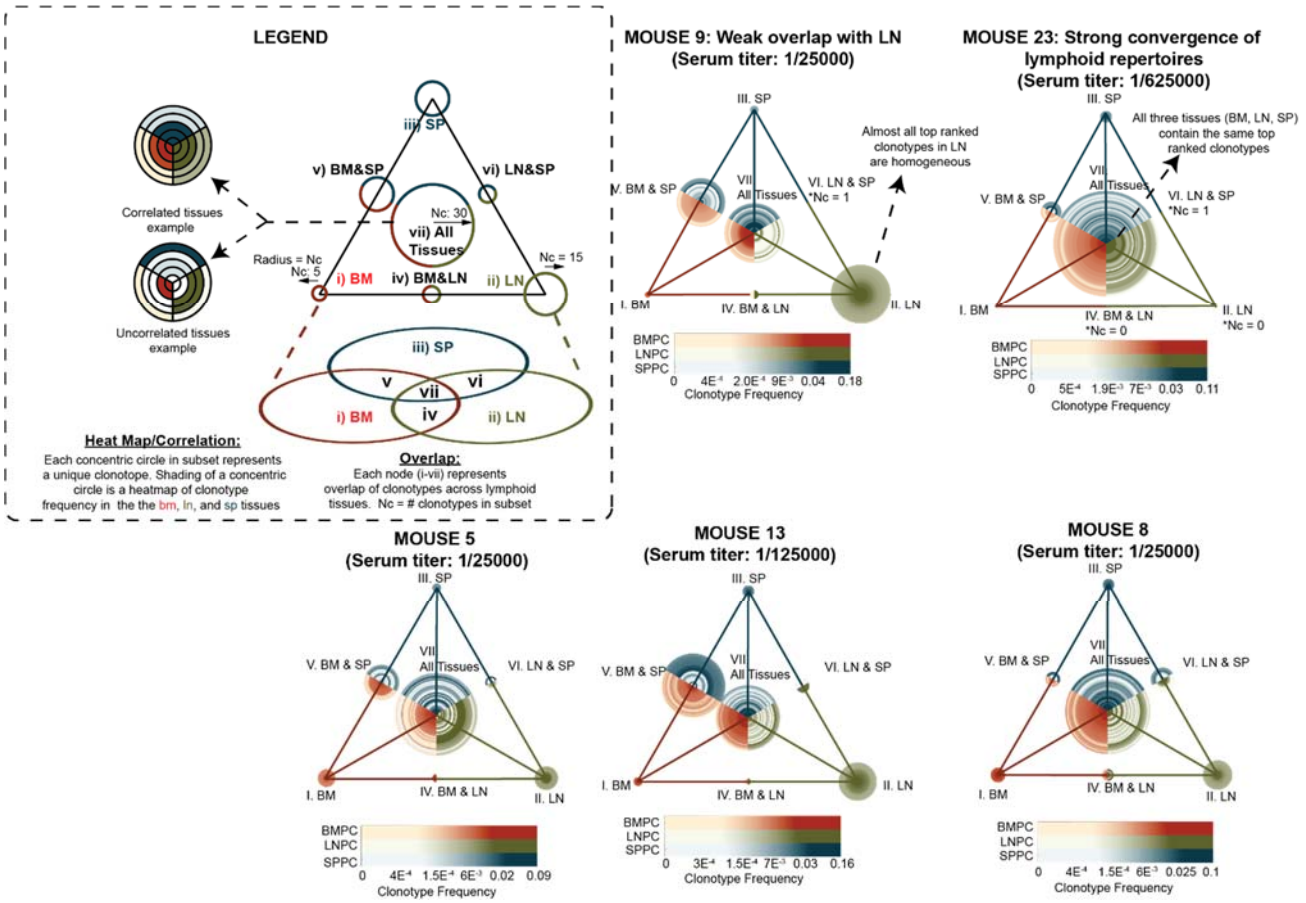


Figure 2.6: Tri-Venn Diagram of lymphoid tissue repertoires

Analagously to a venn-digram, this diagram maps the overlap of the top-30 most highly frequent clonotype-clusters across bone marrow (BM), lymph nodes (LN) and spleen (SP) (nodes i-vii). In addition to overlap, each node is plotted as a heatmap of concentric circles that represent a clonotype's frequency across each respective lymphoid tissue.

Figure 2.6 is similar to a Venn-diagram in the sense that each clonotype-cluster is mapped to one of seven possible subsets (represented as nodes): (i) clonotypes only found in the bone marrow, (ii) clonotypes only found in the lymph nodes, (iii) clonotypes only found in the spleen, (iv) clonotypes found in both the bone marrow and lymph nodes but not spleen, (v) clonotypes found in the bone marrow and spleen but not lymph nodes, (vi) clonotypes found in the lymph nodes and spleen but not bone marrow, and (vii) clonotypes found in all tissues. Each clonotype within a particular node is plotted as a concentric circle, and thus the total number of clonotypes (Nc) within each node is linearly proportional to the radius of the corresponding node. In addition, the color and shading of each concentric circle is proportional to its respective clonotype frequency in a lymphoid tissue. The shade of red, green, and blue for each concentric circle reflects that clonotype's frequency in the bone marrow, lymph nodes, and spleen respectively. Most importantly, concentric circles, comprising node vii, that are colored equally by dark red, dark green, and dark blue lines represents a clonotype cluster found at high frequency in all lymphoid tissues. Hence, each node in Figure 2.6 can be described as a heatmap of clonotype-cluster frequency for a particular subset of a Venn diagram.

Figure 2.6 illustrates that, unlike the other four animals examined here, the V_H repertoire of $CD138^+$ antibody secreting cells in mouse 23, which displayed a significantly higher titer to HEL (1:625000) than any other mouse, was very similar among the three lymphoid organs examined. In this animal, the majority of V_H sequences within the repertoire were represented by a few highly prevalent clonotypes that were distributed equally among the bone marrow, lymph nodes, and spleen (Figure 2.6, mouse 23 node vii). Mouse 23 appeared to have a strong immune response that led to a very large expansion of

a limited set of antigen-specific B cells which subsequently distributed across all three tissues.

Isolation of Antigen Specific Antibodies

As mentioned above, the presence of the same highly abundant V_H genes in different tissues suggests that these V_H genes encode antigen-specific antibodies. To examine this hypothesis, we evaluated the most abundant clonotype-clusters shared in all three lymphoid tissues of mouse 23. At the time this study was performed, it was not possible to determine the paired repertoire of V_H and V_L sequences which are encoded by different mRNA strands. Therefore, in order to determine the antigen specificity of the antibodies encoding the V_H genes of interest (i.e. the most abundant V_H clonotype clusters), it was necessary to identify cognate V_L chains that could pair with the V_H sequence of interest to yield functional antibodies. For this purpose, the five V_H sequences were synthesized using the consensus DNA sequences of the top five clonotype-clusters shared between all three lymphoid tissues in mouse 23 (Table B.5). Each V_H gene was paired with a library of potential V_L genes sequenced from the cDNA of mouse 23 Bone marrow $CD138^+$ cells. The resulting five libraries of V_H:V_L combinations were expressed as fragment antigen binding (FAb) proteins in *E. coli*. Colonies expressing different FAbs were picked at random, grown in 96 well plates, and tested for specificity to HEL. Ten 96-well plates were examined for each of the 5 libraries.

ELISA showed that three of the five synthetic V_H gene candidates had at least four-fold signal above BSA background binding, suggesting specificity towards the HEL antigen. It should be noted that our inability to identify antigen specific antibodies for 2/5 highly abundant VH genes does not necessarily imply that these V_H genes do not encode antigen specific proteins. It is possible, for example, that a cognate V_L gene resulting in a

functional antibody could not be found via screening of a small number of VH:VL combinations. Using competitive ELISA, we analyzed the binding equilibrium between HEL and the three synthetic Fab candidates. Specifically, each Fab was incubated with 80 μg/mL of soluble HEL prior to ELISA analysis. After incubation, the plate was washed such that any FAb bound to the HEL in solution at equilibrium would be washed away. For two candidate FAb genes, competitive ELISA resulted in a more than 48% loss in signal due to the fraction of FAb that was bound to soluble HEL and subsequently washed. Notably, the third candidate (V_H gene variant 2_1), which initially showed a 70 fold increase in signal above BSA binding, resulted in a 70% loss of signal after competitive ELISA (Table 2.5). Therefore, we show that the three most highly abundant V_H clonotype-clusters in all lymphoid tissues of mouse 23 show strong binding specificity against the immunized antigen, Hen-egg-lysozyme.

Unique CDRH3	**Clonotype Rank**			**V_H Gene Variant**	**Signal above Background**	**%reduction competitive ELISA**
	BMPC	LNPC	SPPC			
AREYGGRGFDY	1	1	1	1_1	4 x	-48%
ARDSSGGFAY	2	3	4	2_1	70 x	-70%
ARGGYEGY	5	2	8	3_1	4 x	-67%
ARYGNYEGYAMDY	3	4	2			
AKGPYDYFAY	4	5	11			

Table 2.5: HEL specific antibodies identified from NGS of mouse 23 lymphoid repertoire

DISCUSSION

Advancements in next-generation sequencing technology have revolutionized the field of genomics[107]. NextGen sequencing has also been instrumental in the determination of the repertoire of immune receptors encoded by B and T cells[11], [71], [108]. In this study, we sought to examine features of the V_H repertoire encoded by antibody secreting B cells within each of the major spleen, lymph node and bone marrow lymphoid tissues of the mouse model. We generated approximately $2.8x10^5$ full-length V_H gene sequences and have applied a variety of rigorous statistical analysis methods to compare features of these large datasets. We observed that the lymph node V_H repertoires appears to be dominated by a few highly expanded clones. As a result the measurement of first order true diversity in the lymph node repertoire was significantly lower than that of the spleen and bone marrow. It should be noted that we were able to collect approximately 50% of the lymph nodes from each animal. Lymph nodes are difficult to localize anatomically which makes it extremely difficult to exhaustively extract all the lymph nodes from an animal. Because of this caveat we cannot formally rule out the possibility that the lower degree of diversity reported here might be a consequence of under sampling the lymph node compartment. However the possibility that the repertoire of B cells in the remaining lymph nodes was substantially different in a manner that would alter the conclusion above is remote.

Results of our statistical analyses suggest that the V_H repertoires in bone marrow and spleen tissues are highly similar. Pairwise comparison of lymphoid tissues with lymph node V_H repertoire appeared to be variable across different mouse individuals. In mouse 23, however, we find a strong convergence of the V_H repertoires from all three analyzed tissues. Interestingly, mouse 23 was the only mouse that showed a very strong serum titer

against the HEL antigen. We postulate that this association is not a coincidence and that the higher titer is likely to be a consequence of the synergistic contributions from all the antigen-specific B cell clonotypes located in all three lymphoid tissues. This hypothesis was confirmed with the recombinant expression of the highly correlated antibody sequences to show antigen-specific binding with ELISA. It would be informative to investigate whether this convergence is common in most immunized mice that generate a strong immune response against the antigen of interest. A converging repertoire in multiple tissues may be an ideal data-mining set for antigen specific antibody sequences. This study is one of the few studies that demonstrate the use of high throughput sequencing technology to quantify and compare the immune repertoires of mice after immunization. Further study of B cell trafficking will greatly benefit from greater sequencing depth at decreased costs. However, as sequence depth continues to increase, novel analytical methods will be required to process this vast amount of data.

Chapter 3: Rapid germline V_H gene assignment of immunoglobulin sequences using Fast Fourier Transform techniques

INTRODUCTION

Antibody diversity is generated by the recombination of the germline V, D, and J genetic elements followed by somatic hypermutation of the BCR nucleotide sequence. Characterization of these genetic diversification processes is essential for addressing key aims in the study of immunology such as elucidating the true diversity of antibodies generated after antigen challenge. Furthermore, the recognition of anomalies in V(D)J recombination and SHM can be indicative of disease states and of an individual's ability to mount an effective immune response against various antigens. For example, bias in germline gene usage and somatic hypermutation has been linked to the development of multiple types of autoimmune disease and lymphomas [109]–[112]. To answer such questions, the first step in the analysis of immunoglobulin sequences is typically the assignment an antibody sequence to a corresponding V, D, and J germline gene segment, and analysis of mutations introduced in the segments during differentiation.

However, V(D)J assignment is not straightforward for a number of reasons. 1) human antibodies are derived from more than 250 V_H, 20 J_H, and 30 D_H germline genes and alleles [28]. 2) V, D, and J segments display a high degree of sequence homology. 3) The formation of the CDR3 via random nucleotide addition and deletion at the V-D and D-J junction, and the somatic hypermutation at random nucleotide positions along the variable gene segment result in considerable variation of the recombined V, D, and J gene segments. (4) Finally, the incorporation of non-biological sequencing error must be considered. Put differently, the challenge of V(D)J assignment is aligning a highly mutated

sequence to the correct germline gene segment from a large collection of homologous nucleotide sequences.

Numerous algorithms have been developed for immunoglobulin annotation [67], [68], [113]–[115]. The most widely used algorithm for germline V gene annotation is provided by the IMGT High V quest tool[67]. However, this algorithm is closed-source, researchers must submit their requests to an online queue, and it restricts analysis to only 500,000 sequences per request. Using this online queue to analyze very large datasets can often take several days for data to be returned. On the other hand, many open sourced stand-alone algorithms, such as iHMMune Align and Soda2, which depend upon Hidden Markov Models, can be computationally expensive[113], [114]. Because of the high computational expense, these algorithms are not suitable for processing the amount of data generated by using next generation sequencing (NGS) technologies[116]. Currently, next generation sequencers can produce more than 10^6 reads per sequence run; this throughput continues to grow at an exponential rate. Thus there is a need for more efficient open source tools for germline V gene annotation of very large datasets[117].

Here, we took advantage of Fast Fourier Transform algorithms to develop a method to rapidly perform gapless alignments of immunoglobulin sequences to a database of all possible V_H germline genes. Similar to the FASTA alignment algorithm [118], the initial gapless alignment step identifies the best regions of similarity between two sequences without allowing for insertions or deletions. However, in contrast to a FASTA or BLAST alignment, this method is not restricted to finding regions that have an exact (or highly similar) match between both sequences. In the event that multiple regions of similarity are found (indicative of gaps within an alignment), a modified version of a Smith Waterman

local alignment[119] is used to resolve any insertions or deletions between pairwise sequences.

While this algorithm focusses on human V_H gene assignment, the methods described can be applied to any other set of somatically encoded genes. We show that the algorithm can align more than 10^6 immunoglobulin sequences to their respective germline genes in less than 3 hours on an average desktop computer. Most importantly, comparison to IMGT analysis demonstrates that this method was able to accurately annotate the V_H germline genes of immunoglobulin sequences isolated from NGS immunoglobulin repertoire datasets. This tool for rapidly aligning immunoglobulin sequences to their respective germline genes could be an important advance in antibody repertoire sequence analysis.

METHODS

Hardware and software

All analysis was performed on a standard desktop computer operated by an intel i5 core processor and 8 GB RAM. The IgBlast stand-alone program [68] was installed on a Linux operating system. The modified Smith-Waterman alignment algorithm was first written in Matlab, and then rewritten in c++. The Fourier Germline Assignment was written using Matlab operated by a Windows operating system. The Fourier Germline Assignment program used both the Matlab and the c++ versions of the modified Smith-Waterman alignment. Matlab uses the FFTW3 software package (written in c) for Fast Fourier Transformations [120]. IMGT analysis was performed using the web-based IMGT High V-Quest tool [67]. Matlab was used to compare the results of V_H Gene assignments between IMGT, IgBlast, and Fourier Gene Assignment.

Gapless alignment description and design

The goal of immunoglobulin annotation is to identify the germline V_H gene that aligns best to the sequence of interest. One of the simplest methods for aligning two sequences is a gapless alignment. During gapless alignment, a target sequence (i.e. V_H germline gene) is shifted translationally along the query sequence (i.e. V_H sequence read). The alignment between the two sequences is calculated at each translational shift along the query (Figure 3.1).

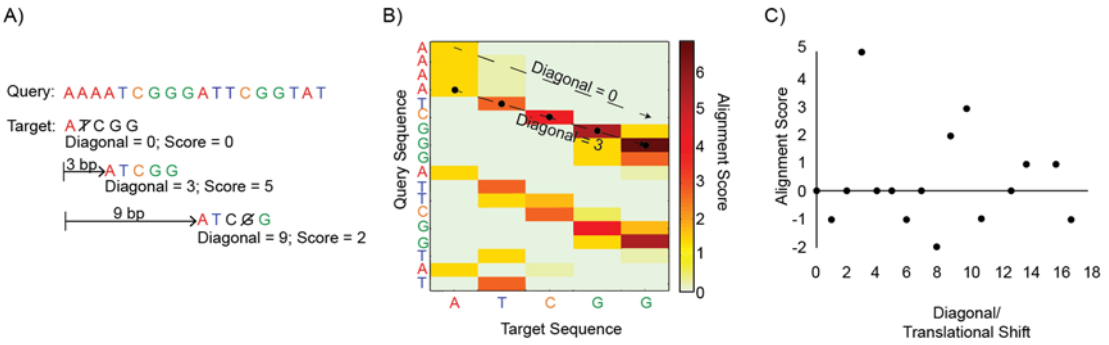


Figure 3.1: Gapless alignment between a target and query sequence

A) Alignment between a target and query sequence is calculated at each position along query. Matching nucleotides are scored by +1, mismatched complementary nucleotides are penalized by -1. The shift of the target along the query is referred to as the diagonal. B) Score-matrix of gapless alignment. C) Scatter plot of gapless alignment as a function of the diagonal

In Figure 3.1, the alignment score between the sequence pair is calculated by scoring matching base-pairs between the query and target sequence by 1 (m=1) and penalizing complementary base-pair mismatches (i.e. A-T) by a penalty of -1 (mm = 1) (Equation 3.1). The position along the query sequence (translational shift) that results in the highest alignment score is considered to be the best region of similarity between the target and query sequence.

$$S(d) = \sum_{i=0}^{N_{SG}-1} Al(\mathrm{i}, d) \Rightarrow Al(\mathrm{i}, d) = \left\{ \begin{array}{ll} +m & if \quad s(i) = g(i-d) \\ -mm & if \quad s(i) = \mathrm{complement}(g(i-d)) \end{array} \right\} [Eq 3.1]$$

Where:

S = Score at that diagonal (translational shift between target and query);

d = diagonal (alignment shift between query (s) & target (g) nt positions);

N_{SG} = combined lengths of query and target sequences;

i = specific nucleotide position within query sequence;

i+d = specific nucleotide position within target sequence;

m = score for matching nucleotides

mm = score for mismatching complementary nucleotides

Al = match/mismatch score at that position

The alignment scoring scheme of Equation 3.1 results in a maximum alignment between the query sequence and target sequence at a "diagonal" of 3 where the 4th nucleotide of the query sequence is aligned to the 1st base of the target. Graphically, this alignment can be illustrated using a scoring matrix (Figure 3.1B). The rows of the matrix correspond to the nucleotide bases of the query sequence whereas the columns correspond to the nucleotide bases of the target sequence. Finally, another method for visualizing gapless alignments is a scatter plot which plots alignment score between two sequences as a function (Equation 3.1) of the current diagonal between the query and target sequence (Figure 3.1C). The gapless local alignment of full length sequences requires a significant number of computational operations. For example, assuming that the query and target sequences are the same length, N, then N^2 nucleotide comparisons are required to identify the best alignment. Thus, this method is rarely considered a computationally efficient method for pairwise sequence alignments.

Gapless alignment in Fourier space

Because local and global alignments using dynamic programming are impractical for aligning many sequences, heuristic algorithms, such as BLAST, which balance both speed and accuracy of nucleotide alignment [118], [121], have been developed and are widely used. One such method takes advantage of applying Fast Fourier Transforms to nucleotide alignment[122]–[124]. While, Fourier transform techniques have been for DNA sequence alignment, they have never been applied to immunoglobulin V_H gene assignment. The simplest calculation of the gapless alignment at every diagonal requires

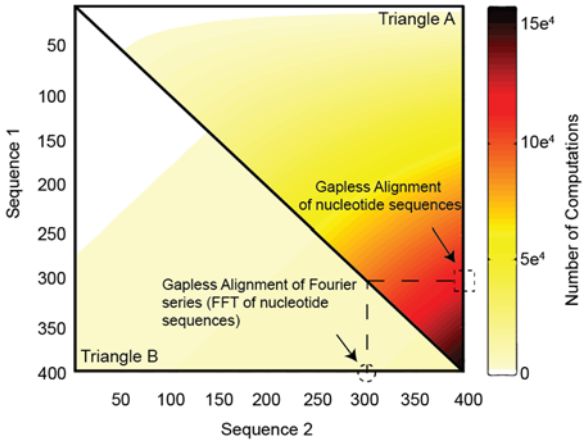


Figure 3.2: Comparison of gapless nucleotide alignment using Fast Fourier Transform

> Top right section of the square (triangle A) represents the number of computations required for nucleotide alignment as a function of sequence length ($O(N^2)$). The bottom left section (triangle B) shows the same operation using FFT (O(NlogN)). The dotted circle and square represent the average alignment of an immunoglobulin sequence (400 bp) to a V_H germline sequence (300 bp).

N^2 operations. However, representing DNA sequences as Fourier series can perform a gapless alignment between two sequences that scales by only NLog(N) operations (Figure 3.2).

As discussed above, Equation 3.1 describes the gapless local alignment score (S) as a function of the diagonal (d) between pairwise sequences. The conversion of each

nucleotide (A,T,C,G) into a system of complex integers (1j,-1j,1,-1)[123] enables a variation of Equation 3.1 to be rewritten as the dot product between the complex integer representation of the query sequence and the conjugate of the complex integer representation of the target sequence (Equation 3.2). Using this scheme to represent DNA sequences, the alignment of matching nucleotides at each position along a diagonal (i.e. A-A) are given a score of 1, alignment of complementary bases (i.e. A-T) are given a penalty score of -1, and alignment of non-complementary bases (i.e. A-C) are not scored. Variations of this scheme can be used to weight nucleotide matches and mismatches similarly (Appendix C). Equation 3.2 shows that the calculation of the alignment score (Sc(d)) at every diagonal, d, is simply the discrete cross-correlation between two complex functions, s and g.

Let A = 0+1j, T = 0-1j, C = 1+0j, G = 1-0j, Then

$$Sc(d) = R\{s \lozenge g\} = \mathrm{R}\{ \sum_{i=0}^{i=N_{sg}-1} s(i) \bullet g^{*}(i-d) \} [Eq3.2]$$

Where

◊ denotes the cross correlation between s and g;

∗ denotes the complex conjugate;

(i-d) represents the diagonal shift (d) between the query and target alignment

$R\{\}$ denotes taking only the real coefficients resulting from the sum

Interestingly, according to the cross-correlation and convolution theorem, the Fourier transformation of the cross-correlation between two functions is equal to the point-wise product of each function's Fourier series coefficients (derivation in Appendix C)[125]. In other words, rather than requiring N^2 operations as shown in equation 3.2, the Fourier transform of a gapless alignment at each diagonal of pairwise DNA sequences can

be calculated using only N multiplication operations in the Frequency domain (Equation 3.3).

Let F[] represent the fourier transform of a function, Then

$$F[s \lozenge g] = \mathrm{F[Sc]} = F[\mathrm{s}] \bullet F[g] \quad [Eq 3.3]$$

Additionally, using the Fast Fourier Transform (FFT), both discrete functions can be transformed into their respective Fourier series in NLogN operations[126]. Thus, by representing each nucleotide sequence as a discrete complex function (A=1j,T=-1j,C=1,G=-1), the combination of the Fast Fourier Transform with the cross-correlation theorem can be used for the accurate gapless alignment between two DNA sequences within an order of NlogN operations (Table 3.1).

Step	DNA IgSequence	Germline Gene Segment	# Operations	Notes
1	Convert DNA to complex integer series	Convert DNA to complex integer series	N	N: Sequence length A =0+1j, T = 0-1j, C=1, G=-1
2	Transform series using FFT	Transform series using FFT	2NLogN	
3		Convert to complex conjugate	N	
4	Point-wise multiplication between each Fourier series coefficient		N	Fourier of cross-correlation (Eq 3.3)

Table 3.1 Steps for performing gapless sequence alignments using FFT

5	Take the inverse of the resulting series using FFT. Considering only the real coefficients.	NLogN	Real coefficients of inverse transform represents the gapless sequence alignment at that diagonal (Eq 3.2)
Total	Performs gapless alignment of pairwise sequences	O(NLogN)	Match(A-A): +1 Comp mismatch(A-T): -1 Other mismatch(A-C): 0

Table 3.1 continued

Alignment score parameters

The alignment between pairwise sequences was calculated using four parameters. (1) The *match score* is a positive value specifying the score assigned to matching nucleotides in a pairwise alignment. (2) The *mismatch score* is a negative value specifying the penalty for aligning mismatching nucleotides. (3) The *gap-open penalty* specifies the negative value for introducing an insertion or deletion (in-dels) in a pairwise alignment. (4) The *gap-extension penalty* specifies the negative value for extending insertions or deletions using affine gaps. Affine gap penalties are used when there are consecutive insertions or deletions in an alignment. Table 3.2 summarizes the values assigned to these parameters when using Fourier alignments and when using local Smith-Waterman (SW) alignment.

Alignment Score Parameter	Fourier Alignment	Smith-Waterman alignment
Match score	1	5
Mismatch score: Non-complementary bases (i.e A-C, T-G)	0	-4
Mismatch score: Complimentary bases (i.e. A-T, C-G)	-1	-4
Gap open penalty	-10	-20
Gap-extension penalty	-2	10

Table 3.2: Parameters used for pairwise alignments

Clustering V_H germline genes

In the germline assignment algorithm discussed in the results below, the set of V_H germline genes are first grouped into clusters based on nucleotide sequence identity. In this regard, we first calculated the pairwise alignment for all 257 human V_H germline genes (downloaded from the IMGT germline database (Giudicelli et al., 2005)) using Smith-Waterman (SW) local alignment. The scoring parameters of the alignment are defined above in Table 3.2. Agglomerative average-linkage hierarchical clustering (UPGMA[127]) of the V_H genes was performed using the Euclidean distance between pairwise alignment scores. From the hierarchical cluster, we selected a threshold cutoff

that clustered of V_H germline genes with at least 90% pairwise nucleotide identity (Figure 3.3).

Each cluster is represented by a consensus nucleotide sequence derived from the multiple alignment of every V_H gene within the cluster. For each cluster, we performed a second Smith-Waterman pairwise in which every V_H gene was aligned to the nucleotide consensus sequence. Any V_H gene that contained an insertion or deletion when aligned to the consensus cluster sequence was removed from the cluster and placed into a new cluster group. This process was repeated until every V_H gene grouped into a cluster did not have

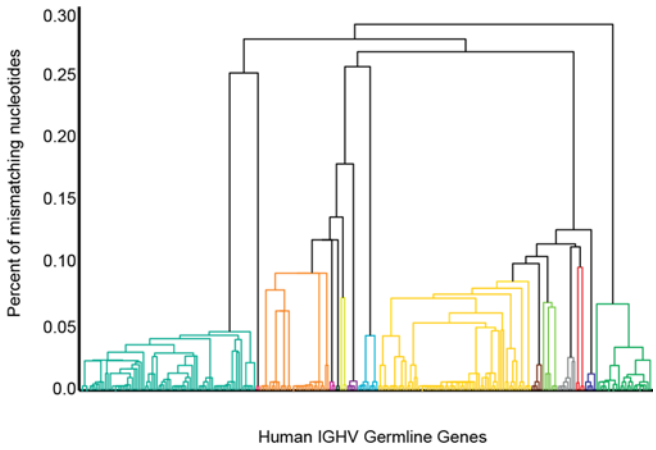


Figure 3.3: Hierarchical clustering of IGHV germline.

Human germline V_H genes were hierarchically clustered using the Smith-Waterman pairwise alignment score as a distance matrix. The figure shows the results of the hierarchical clustering. Branches of the tree are colored by V_H sequences that have more than 90% sequence identity to one another.

any insertions or deletions when aligned to its respective cluster consensus sequence. In total, all 257 V_H germline genes were clustered into 19 groups containing germline genes

with 90% pairwise similarity and no insertions or deletions relative to the consensus sequence (Table C.1).

RESULTS

Proof of concept: immunoglobulin sequence alignment using Fourier transforms

Somatic hypermutation, a biological process in which random nucleotide substitutions are introduced into an immunoglobulin sequence, is the principle source of variation between a full length immunoglobulin sequence and its corresponding germline gene. Insertions and deletions are rarely introduced by somatic hypermutation and it was reported that less than 10% of circulating B cells encode antibodies with identifiable insertion or deletion mutations (in-dels) when aligned to their respective germline genes [128], [129]. Thus, the use of gapless alignment techniques using the Fast Fourier Transformation posed a promising method for accurately identifying the correct V_H germline gene from a known set of full length gene segments (Figure 3.4).

First, we wanted to show that gapless alignment is an accurate method for annotating immunoglobulin sequences. As a proof of concept, we tested the effectiveness

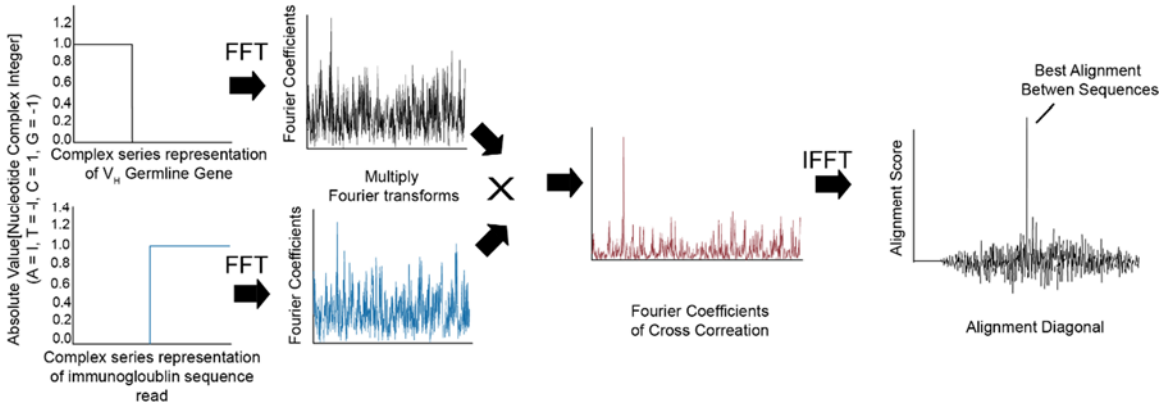


Figure 3.4: Gapless alignment of immunoglobulin sequence to germline using FFT

of the algorithm described in Table 3.1 using NGS data from both a mouse repertoire study (mouse_test_10k; 10,000 reads total) and from a human repertoire study (hum_testset_10k; 10,000 reads total). First the respective database of all V_H germline genes was downloaded from the IMGT germline database [130]. At the time this work was performed, the database included 257 unique alleles for the humans and 350 alleles for mice (IMGT version 3.0). In Equations 3.2 and 3.3, each real coefficient resulting from the discrete inverse Fourier transform represents the alignment score at different diagonals between the pairwise sequences. The maximum value of real coefficients represents the best possible gapless alignment between the pair; the position of the maximum value represents the diagonal with the best alignment between the sequence read and that specific germline allele. In other words, the position of the maximum alignment corresponds to the most probable position along the sequence read where the V_H germline gene starts.

Each sequence read was assigned to the germline gene that showed the best maximum alignment to either the forward or reverse complement of the sequence read. We then compared the results of a gapless alignment method to gene assignments determined by the IMGT high V-quest analysis [67]. We found that, using the FFT gapless alignment, 9752/10000 (~97%) sequences from the mouse dataset and 9608/10000 (~96%) sequences from human dataset had identical germline-gene assignment as compared to results from the IMGT analysis. Such high similarity with IMGT showed that, without any significant optimization of the algorithm, a FFT based alignment could be an effective and rapid method for gene annotation.

We investigated the sequence reads that were annotated differently from IMGT analysis. We focused our efforts on identifying the differences in the human data set, hum_testset_10k. In total, 392 sequence reads from hum_testset_10k indicated a mismatch

between the two annotation methods (IMGT and FFT). Importantly, the minor discrepancies between these sequences did not bias the total distribution of V_H gene usage reported by IMGT and FFT analysis (Figure 3.5A). The absence of a systematic bias suggests that the FFT alignment algorithm is able to identify sequence reads derived from all possible V_H genes in the database. Instead, inaccuracy in the assignment was most likely due to unexpected and infrequent mutations in the V(D)J rearranged immunoglobulin sequence read.

Smith-Waterman (SW) local gapped alignment was used to determine the correct germline gene assignment for these 392 sequence reads. Using Smith-Waterman, each problematic sequence read was aligned to both of the predicted "best-aligning" germline gene annotated by IMGT and the FFT alignment method. The correct germline call was then assigned to the gene segment with the highest gapped alignment score determined by Smith-Waterman. For the majority of mismatch calls (278/392), alignment to either of the two predicted germline genes had the same Smith-Waterman alignment score. That is, the sequence read aligned equally well to both predicted gene segment sequences and, thus, it was not possible to determine which gene annotation was correct. Furthermore, Smith-Waterman alignment indicated that, for a very select number of sequences (21/392), the FFT predicted germline gene had a better alignment score to the sequence read as compared to the IMGT predicted germline gene. Further inspection revealed that this discrepancy was most likely a result of IMGT annotation showing bias for gene alignments with insertion and deletion mutations in the CDR region as compared to nucleotide mismatch errors. For example, in one specific situation, IMGT predicted that the best germline gene for a sequence read was IGHV3-NL1, whereas the other alignment methods (FFT and SW) predicted that the best germline gene was IGHV3-66. In this example, alignment of

IGHV3-NL1 with the sequence read resulted in the deletion of three nucleotides in the CDR2 region, while, alignment of the sequence with IGHV3-66 did not contain in-del mutations but instead had an additional 5 base-pair mutations in the CDR1 region (Figure C.1). We did not further investigate these rare occasions in which the true gene assignment was unclear.

Smith-Waterman analysis supported IMGT annotation for the remaining 92 sequences. We looked at the most common types of annotation errors identified in these 92 sequences. Figure 3.5 B plots the distribution of genes incorrectly assigned by the FFT gapless alignment method. Each row of the matrix represents a germline gene correctly

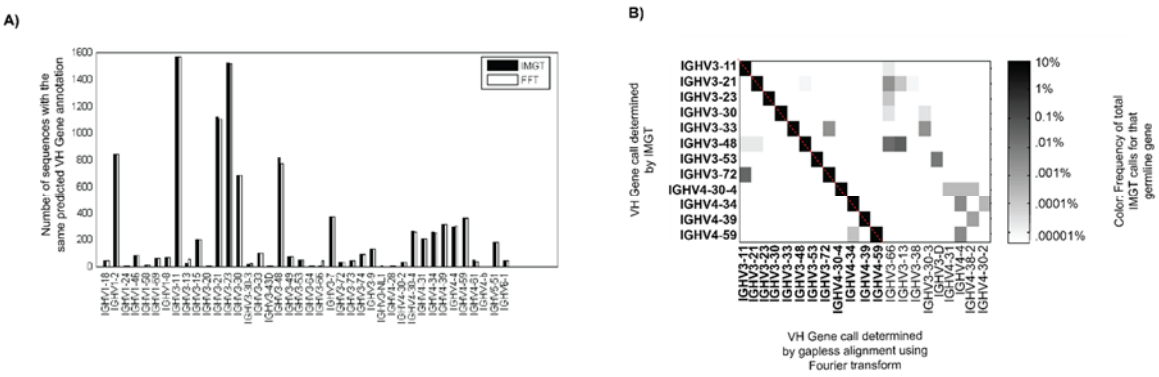


Figure 3.5: Comparison of IMGT V_H annotation to FFT gapless alignment annotation

> A) Distribution of V_H gene usage in hum_testset_10k. Black bars represent V_H genes annotated by IMGT. White bars represent V_H genes annotated by gapless FFT alignment method. B) V_H genes annotated incorrectly by gapless FFT alignment. Rows represent the V_H gene annotation determined by IMGT. Columns represent the FFT assignment for that V_H gene. Color of boxes represents the frequency each specific inaccurate assignment. The boxes along the dashed-red diagonal line represent sequences correctly annotated by both algorithms. The three most miscalled assignments was IGHV3-48 mistaken for IGHV3-66 and IGHV3-13, and IGHV3-72 mistaken for IGHV3-11

identified by IMGT but not by FFT; for each row, each column represents the resulting

gene predicted by gapless alignment. It is evident from Figure 3.5B that no germline V_H gene is consistently annotated incorrectly. IGHV3-48 was the most commonly miss-annotated germline gene where, for approximately 5% of the sequences in the dataset, sequences supposedly derived from IGHV3-48 were assigned to either IGHV3-66 or IGHV3-13. Smith-Waterman analysis revealed that inaccurate assignment of IGHV3-66 or IGHV3-13 was due to the inability to account for insertions and deletions between pairwise sequences. When accounting for insertions and deletions, IGHV3-48 was a better alignment to the sequence read than either IGHV3-66 or IGHV3-13 (Figures C.2 and C.3).

Insertion-Deletion correction using a variation of the Smith-Waterman local alignment algorithm

The above FFT alignment method was very effective at locating the position or diagonal between pairwise sequences that resulted in the best gapless alignment. However, using only the maximum alignment along this diagonal will not yield the maximum global alignment and may result in potential errors when annotating immunoglobulin sequences containing insertion and deletion mutations. Figure 3.6 illustrates this potential drawback to using gapless alignments. In figure 3.6, an immunoglobulin sequence from a NextGen experiment is aligned to both IGHV3-66 (Figure 3.6A) and IGHV3-48 (Figure 3.6B) using gapless alignments. The position resulting in the maximum alignment between both germline genes is at diagonal 11; at that diagonal, however, the germline gene IGHV3-66 shows a better alignment to the read with a max alignment score of 252 as compared to IGHV3-48 with a max alignment score of 160. This result differs from a Smith-Waterman alignment in which IGHV3-48 shows a better alignment. The SW local alignment indicated that the optimal alignment path includes a deletion of 3 base pairs in the sequence

to the IGHV3-48 germline (Figure C.3). This deletion is depicted in Figure 3.6B where gapless alignment results in a second non-random alignment at diagonal 8.

The detection of multiple non-random nucleotide alignments is an indication that insertions or deletions must be present in the pairwise alignment. Therefore, we used the position of each non-random alignment (defined in Appendix C) to guide a more accurate

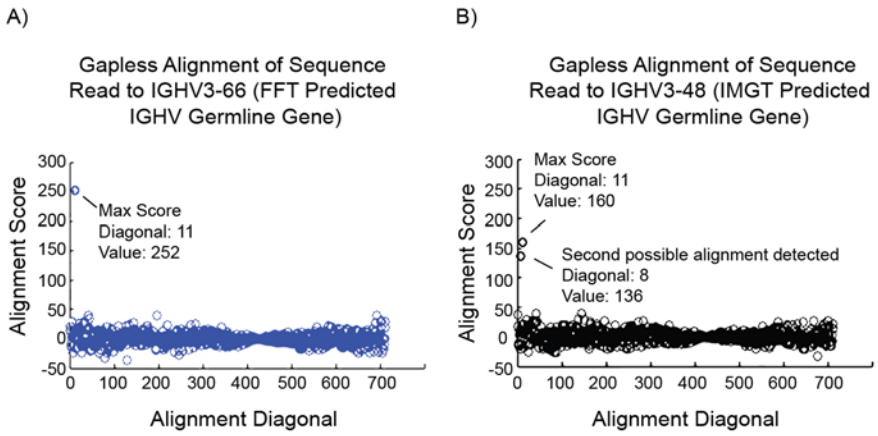


Figure 3.6: FFT alignment of a NextGen sequence read to two possible germline gene sequences

A) Scatter plot of gapless alignment to IGHV3-66. Alignment to this germline does not have insertions or deletions and results in only one maximum alignment score of 252. B) Scatter plot of gapless alignment to IGHV3-48. SW optimal alignment to this germline includes a deletion of 3 base-pairs in the sequence. As a result, two peaks above background are detected at diagonals 8 and 11. The maximum alignment, 160, is lower than the alignment to IGHV3-66. However, summation of both alignments at positions 8 and 11 would results in a higher alignment score.

algorithm that could account for gaps between sequences. Rather than using the canonical smith-waterman alignment that compares all N^2 space between pairwise sequences, we only considered the alignment space between nucleotides within a predefined distance from the maximum alignment diagonal. This diagonal was identified from the position of the

maximum peak in a gapless alignment, and the maximum allowed search distance from the diagonal is determined by the distance between additional non-random alignment scores. Figure 3.7 illustrates how this method can be applied to the alignment of the test sequence to IGHV3-48 in which two non-random scores, separated by three base-pairs, were detected. Starting at the maximum diagonal, 11, and considering only alignment paths separated by 3 base-pairs, the optimal alignment path between the sequences was found in (3+1)*N steps (Figure 3.7B). For comparison, using dynamic programming where all N^2 alignment paths are considered, results in the same optimal alignment path between the sequences (Figure 3.7 A). These initial benchmarks and analyses show that gapless alignment can be used to accurately align heavy chain sequences to the germline. Therefore, these methods were combined into one algorithm for germline gene assignment.

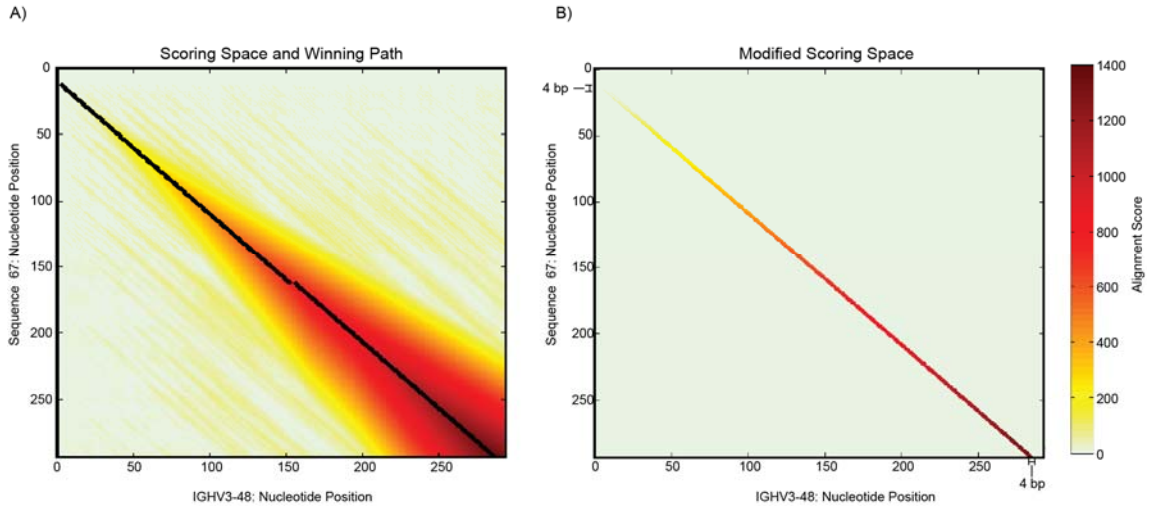
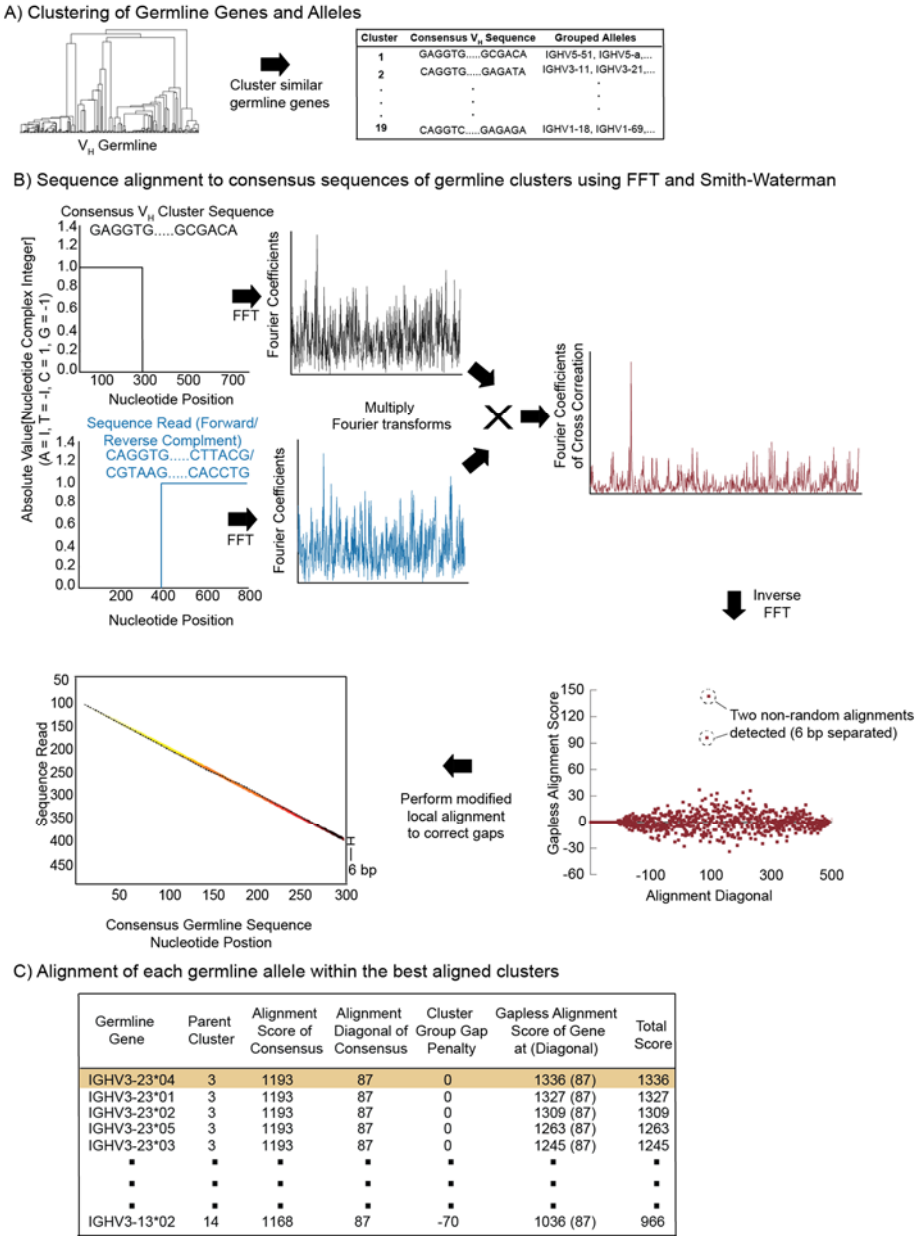


Figure 3.7: Dot plot illustration of the modified local Smith-Waterman alignment

A) Scoring matrix resulting from the canonical SW. SW considers all possible alignment paths between sequences B) Modified scoring matrix. Alignment starts at maximum diagonal alignment position 11, and only considers paths that are separated by an additional 3 base pairs from this maximum diagonal 11. Both methods can identify the 3 base pair deletion in the sequence read.

Germline assignment algorithm



Germline Gene	Parent Cluster	Alignment Score of Consensus	Alignment Diagonal of Consensus	Cluster Group Gap Penalty	Gapless Alignment Score of Gene at (Diagonal)	Total Score
IGHV3-23*04	3	1193	87	0	1336 (87)	1336
IGHV3-23*01	3	1193	87	0	1327 (87)	1327
IGHV3-23*02	3	1193	87	0	1309 (87)	1309
IGHV3-23*05	3	1193	87	0	1263 (87)	1263
IGHV3-23*03	3	1193	87	0	1245 (87)	1245
▪	▪	▪	▪	▪	▪	▪
▪	▪	▪	▪	▪	▪	▪
▪	▪	▪	▪	▪	▪	▪
IGHV3-13*02	14	1168	87	-70	1036 (87)	966

Figure 3.8: Fourier Germline Assignment Algorithm

A) VH germline genes are clustered by pairwise nucleotide similarity. B) Each cluster consensus sequence is aligned to immunoglobulin sequence using FFT. In-del mutations are fixed using the modified SW method. C) The germline genes best aligning clusters are aligned to sequence at the identified maximum alignment diagonal. The gene with the best alignment score is assigned.

The algorithm for annotating the IGHV genes of human immunoglobulin repertoires is organized into three principal steps (Figure 3.8). First, the set of V_H germline genes are grouped into clusters based on pairwise identity. Second, each sequence read is aligned to each consensus cluster sequence using the Fourier series gapless alignment method. The optimal alignment path in sequences which contain insertions and deletions when compared to the cluster sequence is determined using the modified Smith-Waterman gapless alignment. Finally, after selection of clusters which best align to the germline sequence, each V_H gene segment within the cluster is aligned to the sequence at the proper diagonal.

Clustering V_H germline genes

The goal of Fourier analysis for each sequence read was to (1) identify the position (alignment diagonal) where the immunoglobulin sequence begins relative to a germline gene, and (2) determine whether the modified smith-waterman method would be needed to optimize the alignment path in regions containing insertions and deletions. Accounting for these objectives and given the intrinsic nucleotide similarity between germline genes, performing Fourier analysis on every germline gene segment is unnecessary. For example, when aligning a sequence read to the germline, the diagonal resulting in a maximum gapless alignment does not vary significantly between most germline genes (Figure 3.9). In this example illustrated by figure 3.9, more than half of the germline genes aligned to the NextGen sequence show a maximum alignment at diagonal 11. We only need to perform the FFT alignment step on a select number of germline genes to predict this maximum alignment position. Therefore, as described in the methods, we grouped together V_H germline genes with more than 90% pairwise identity at the nucleotide level and no indications of in-del mutations with respect to other genes within the cluster. Based on this

criteria, all 257 germline genes were grouped into 19 clusters, and a consensus nucleotide sequence was determined for each cluster.

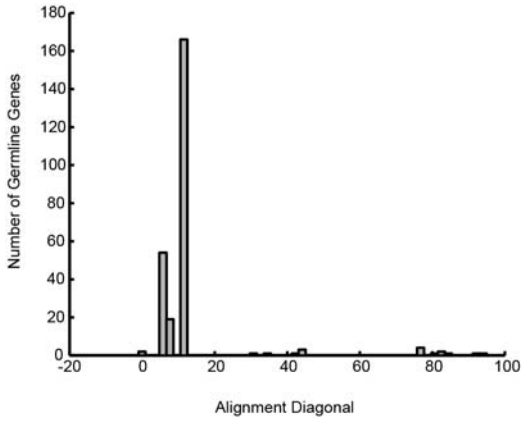


Figure 3.9: Diagonal of maximum alignment between NextGen sequence and V_H germline

Distribution of the predicted starting alignment position of each germline gene aligned to the same immunoglobulin sequence.

Fourier alignment to consensus sequences

As outlined in Figure 3.8B, each sequence read was aligned to the consensus sequences of the individual 19 clusters. The forward and reverse complement of each sequence read were aligned to each cluster consensus sequence using the Fast Fourier Transform. The direction, forward or reverse complement, containing the immunoglobulin sequence was determined by the summation of the maximum alignments in every cluster. We selected the direction (forward or reverse complement) which had the highest alignment score to each cluster. The diagonal of maximum alignment in the determined direction was recorded for each cluster. Additional peaks near the maximum diagonal were identified based on having an alignment score at least three standard deviations above the expected binomial distribution of alignment score between random nucleotide sequences

(Appendix C). Finally, the presence of insertions and deletions near the ends of the sequence read, which would be below the sensitivity of the Fourier analysis, were also determined.

For sequences containing no identifiable insertions or deletions, a final cluster alignment score is calculated where all base pair matches at the best alignment diagonal are given a score of +5, and all base pair mismatches are given a penalty of -4. The modified Smith-Waterman method is applied to alignments predicted to have in-del mutations. For gapped alignments, gap openings are penalized with a score of -20 and gap extensions are penalized with a score of -10; similarly to gapless alignments, base pair matches are scored +5 and base pair mismatches are penalized by -4. The output of the gapped alignment reports both an alignment score and a "gap-corrected" immunoglobulin sequence read where insertions and deletions to the cluster have been removed. After alignment to each consensus sequence, only clusters containing an alignment score within 85% of the maximum cluster alignment score are selected for further analysis.

Final alignment to germline clusters

In the preceding step, the Fourier and SW analysis report (1) whether the NextGen sequence read refers to the forward or reverse complement of an immunoglobulin sequence, (2) a list of cluster consensus sequences that best align to the sequence, (3) the diagonal of best alignment, (4) and the position and number of any insertions or deletions in the alignment of the sequence to the cluster consensus sequences. We can assume that each V_H gene within a cluster will align to the sequence read at the same position as its corresponding cluster consensus sequence. Moreover, because all of V_H germline segments within a cluster do not have any insertion or deletion mutations when aligned to the cluster consensus sequence, we can also assume that each V_H germline segment will

not have insertions or deletions with respect to the "gap-corrected" immunoglobulin sequence read. Using this information, each germline gene within a cluster is aligned to the modified sequence without allowing for any gaps at the diagonal defined by the consensus sequence (Figure 3.8C). A gap penalty is introduced in the finalized alignment scores of germline genes whose consensus sequence had in-del mutations when compared to the original sequence read. Germline genes with the top five best alignment scores to the sequence read are selected as the most likely V_H gene used during VDJ recombination of the heavy chain.

Implementation of the germline assignment algorithm

The germline assignment algorithm was written using Matlab. The effectiveness of the program was first tested on the previously analyzed hum_testset_10k NGS dataset. The results of the alignment was compared to an IgBlast analysis. IgBlast, which relies upon the BLAST alignment algorithm, is a well-validated open-source software tool for annotating immunoglobulin sequence data [68]. Analysis of hum_testset_10k using our germline assignment method took 2 minutes. Gene assignment was found to be 99.5% identical to IMGT annotation (Figure 3.10). More importantly, using the Smith-Waterman method to account for insertions and deletions corrected the initial problems where IGHV3-48 was incorrectly assigned to IGHV3-13 or to IGHV3-66. By comparison, the c++ optimized IgBlast analysis of 10,000 sample sequences took 10.5 minutes. IgBlast analysis also had a V_H gene assignment accuracy greater than 99% as compared to IMGT.

Our germline assignment algorithm, namely the addition of a Smith-Waterman alignment, was designed around correcting for annotation errors we initially noticed when aligning the V_H germline gene segments to hum_testset_10K using only the FFT algorithm. We wanted to test whether we over-trained the algorithm towards the hum_testset_10k.

That is, we wanted to see if, in the process of designing the germline assignment algorithm, we only addressed problems we specifically encountered in the hum_testset_10k. To investigate whether this algorithm was indeed a robust method for annotating V_H NextGen data, we collected another set of 20,000 sequences randomly selected from a second human repertoire project, hum_testset_20k. Analysis of these sequences took 3.5 minutes and again resulted in V_H annotation with greater than 99% accuracy as compared to IMGT analysis. Therefore, we show that this algorithm is a robust method for annotation of the V_H gene segment of an immunoglobulin. Given the rate of analysis for this novel method, it is projected that, using this non-optimized code in Matlab, 1,000,000 sequences can be analyzed, with greater than 99% accuracy, in under 3.5 hours. From previous experience, we project that optimization of this code using c++ would most likely improve this rate by three fold.

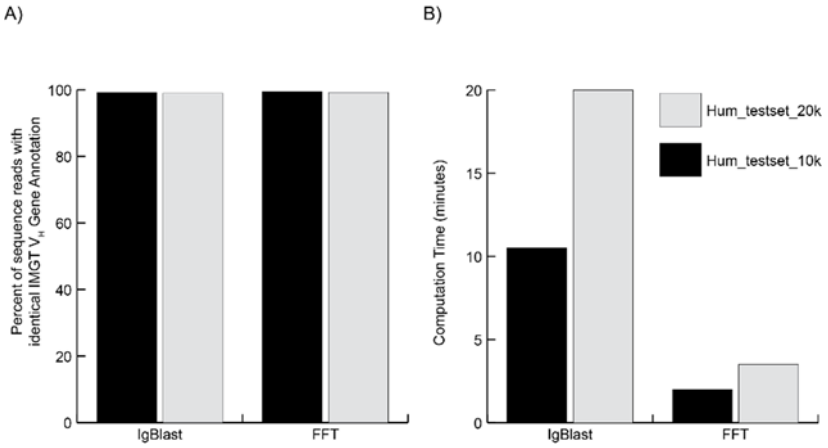


Figure 3.10: Evaluation of the Fourier Germline Assignment algorithm

A minority (5%) of sequences in hum_testset_20k could not be annotated using the IMGT software. These reads most likely represented either non-heavy chain

immunoglobulin sequences or highly mutated sequences that could not be aligned to the germline. We wanted to investigate whether our germline assignment program could also predict these sequences as non-immunoglobulins. For each sequence read, we analyzed the maximum possible gapless alignment score between the cluster consensus sequences. Expectedly, the average score of maximum alignment for these sequence reads, 44, was much lower than the average score of sequence reads with a significant alignment to germline, 253 (Figure 3.11A).

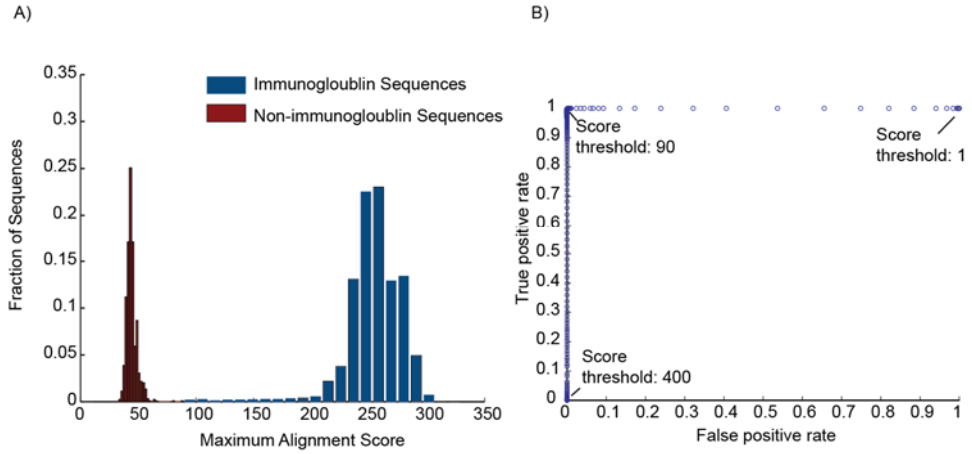


Figure 3.11: Selectivity of the germline assignment algorithm

A) Distribution of the maximum alignment score for non-immunoglobulin sequences (red and immunoglobulin sequences (blue). B) ROC curve demonstrating the selectivity of the algorithm.

Receiver operating characteristic (ROC) analysis of these sequences versus sequences with identifiable V_H germline genes illustrates the performance of the algorithm as a classifier for immunoglobulin sequence reads with a full length V_H gene segment. Figure 3.11B shows that using a cutoff maximum alignment score of 90 can filter out all false positive alignments yet still identify more than 99.8% of the reads with the correct

assignment. In conclusion, we have shown that this method is a very robust method for the rapid annotation of sequence reads to their respective germline.

FUTURE TECHNIQUES: APPLICATION OF SPARSE FFT METHODS

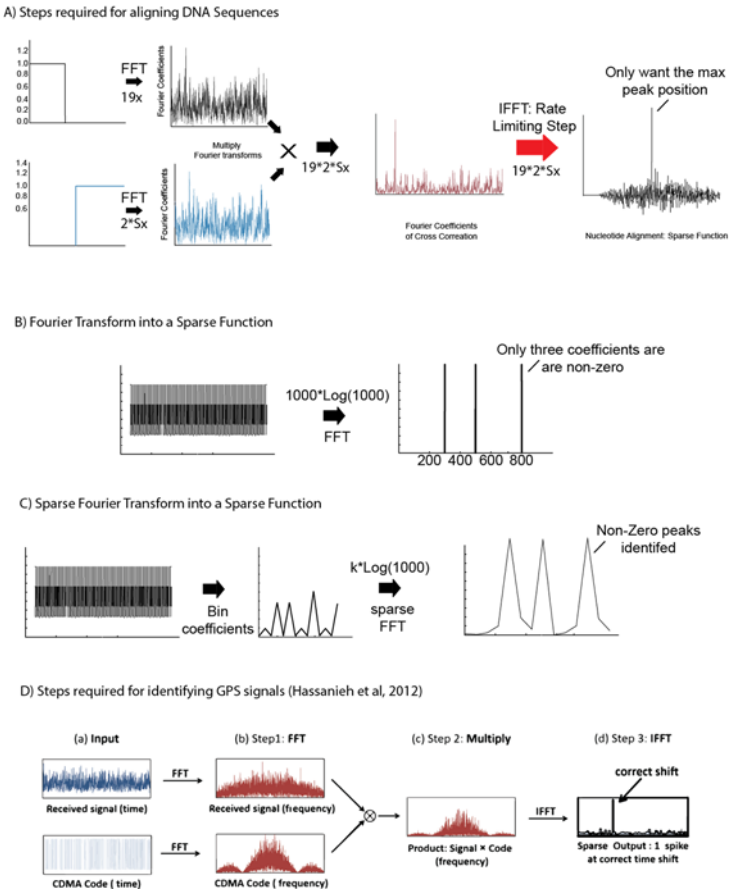


Figure 3.12: Illustration of sparse functions and the sparse FFT

A) Performance of the germline assignment algorithm is limited by the final inverse FFT step (thick red arrow). B) Example of a function whose Fourier series is sparse but required NLogN operations C) Simplified example of a sparse FFT. SparseFFT can accurately estimate the sparse Fourier coefficients in O(klogN) operations. D) Analogous example of the application of a sparse FFT using GPS signaling (Illustration taken from Hassanieh et al., 2012b).

The primary purpose of the Fourier method is to align a query sequence to each potential germline and identify the starting alignment position of the germline along the sequence. One of the rate limiting steps in this effort is the transformation of the cross correlation Fourier series back to the time domain (Figure 3.12A). For example, the alignment of a dataset containing S_x sequences to 19 consensus cluster sequences requires $2*S_x*19$ inverse Fourier transforms. One area of improvement is the ability to enhance the speed of this essential inverse transformation step.

Many novel algorithms have arisen in recent years which further optimize and improve upon the speed of Fourier transforms. One method gaining recently popularity is the development of sparse FFT algorithms [131]. A sparse function can be considered as a large vector of complex integers but only contains K non-zero coefficients (Figure 3.12B). The use of a sparse FFT arises when a non-sparse complex function has a corresponding Fourier series that is sparse and vice-versa.

For example, the original function in in figure 3.12 B has 1000 coefficients or discrete data points, and yet its Fourier transform has only 3 non-zero coefficients. With respect to the canonical Fourier transform function, we have to take the inverse transform of all original 1000 data points to identify the three non-zero Fourier series coefficients (1000*Log(1000) computations are required). The sparse FFT, on the other hand, is designed to estimate those k=3 sparse Fourier coefficients in O(k*Log(N)) computations (Figure 3.12 C) . In summation, given a discrete function (length N) whose corresponding Fourier transform contains K non-zeroes coefficients, the sparse FFT is designed to estimate the value of these K-largest coefficients without taking the Fourier transform on all N data points [131]. With respect to DNA sequence alignments, the cross-correlation or gapless alignment can be considered a sparse function because we are only interested in

identifying the position resulting in the maximum alignment score between pairwise sequences. The remaining "coefficients" in the cross-correlation only correspond to scores caused by alignment of random nucleotide sequences.

Thus, the sparse FFT can be an intriguing method for improving the Fourier germline assignment algorithm. Specifically, we can apply the sparse FFT method in the final rate limiting final step where we take the inverse transformation of the cross-correlation Fourier series. As an example, the application of the sparse FFT method has already been shown to improve the challenge of identifying GPS signals from satellites [132]; this process is highly analogous to our method of gapless nucleotide alignment in Fourier space (Figure 3.12D). Appendix C provides an example algorithm for performing a sparse FFT where we only identify the position of the maximum alignment peak. Figure 3.13 B shows how this technique can be used to identify the maximum alignment position between an immunoglobulin read and its corresponding germline. Application of this algorithm could improve the speed of the inverse transform by tenfold.

Although, a promising technique, there are significant areas for optimization. Sparse FFT algorithms take advantage of a property known as aliasing in which coefficients are binned together before taking the Fourier transform. This method of binning coefficients allows one to take the Fourier transform of a function smaller than N. Binning is not problematic for theoretically sparse functions in which almost every coefficient is zero. However, in the case of nucleotide alignments, the sparse function has many noisy (non-zero) coefficients arising from the random alignment of nucleotide bases. The effect of binning too many coefficients to improve performance will significantly decrease the signal-to-noise ratio of a true alignment from a series of random alignments binned together. Thus, the use of sparse FFT algorithms in nucleotide alignments is limited

by the binomial variance of noise caused from random alignments. In order to fully take advantage of the sparse FFT algorithms, it will be necessary to develop solutions for decreasing the noise of cross-correlation function or improved sparse FFT algorithms which are less sensitive to small non-zero coefficients.

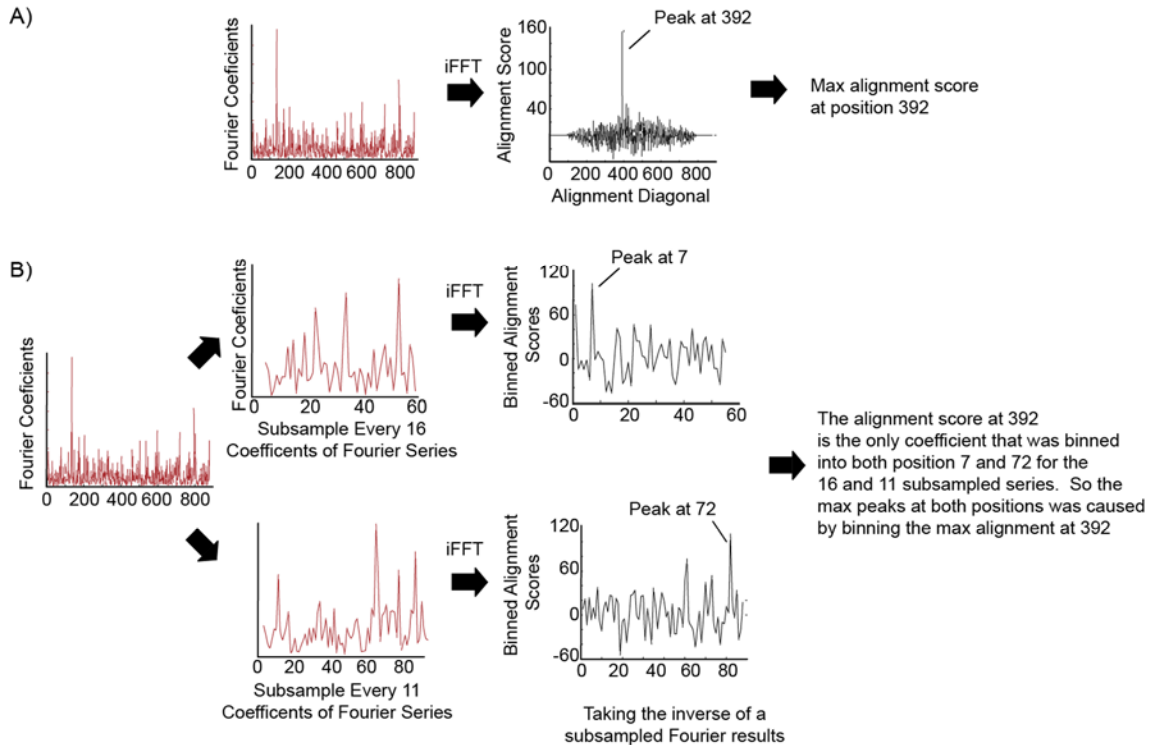


Figure 3.13: Sparse FFT for nucleotide alignments

A) Taking inverse FFT of the cross-correlation Fourier series results in the accurate alignment score at each diagonal. The maximum alignment score is identified at a diagonal 392. B) Taking the inverse of a subsample of the original cross-correlation function results in binning alignment scores from multiple diagonals together. The theory behind the sparse FFT is that binning the maximum alignment score will always result in the maximum peak within the bins. The only alignment score binned into the 7^{th} and 72^{nd} coefficient of the 16-subsampled and 11-subsampled functions, respectively, is the alignment at 392. Maximum alignment can be assumed to be at 392. However, binning together alignment scores significantly decreases the signal-to-noise ratio of a non-random alignment.

CONCLUDING STATEMENTS

We have presented a novel algorithm for the accurate V_H gene assignment of immunoglobulin sequence reads isolated from human B cells. Most importantly, we show that, using a normal desktop computer, this algorithm can analyze data produced by NGS within a few hours. Although we have shown this to be a very promising technique, there are still areas for improvement. Future efforts will focus on adapting this method for the annotation of other species and extensions to J_H gene assignment. Furthermore, the benchmark comparisons indicate that there may still be areas where accuracy can be improved. Finally, optimization of this algorithm using c++ would further enhance performance.

The main advantage of this algorithm is its ability to perform rapid gapless alignment of very long sequences. One limitation of this algorithm is that it was designed to only analyze full length V_H genes from immunoglobulin sequences. For example, this method will not accurately predict events such as gene conversion of V_H genes which have been observed in rabbits [1]. Other alignment algorithms such as BLAST rely instead upon gapless alignments of very small sequences. Consequently, one advantage of the BLAST algorithms, is the ability to both predict gaps in large sequences and predict unexpected recombination events. Gene conversion, however, has not been commonly observed in human and mice repertoires. Thus, this method is suitable for the analysis of most repertoire studies.

In addition, it should be stressed that the primary focus of the algorithm is to rapidly identify a small subset of most likely derived V_H genes from a large database of germline genes. This germline assignment method should act as supplement for downstream algorithms in analysis pipelines which can align immunoglobulin sequences to the

identified germline gene using accurate models which can predict hot spots mutations and experimental error. In conclusion, we have shown that this method is indeed a powerful technique for gene assignment and that it can continue to be improved especially in the application of novel transforms such as the sparse FFT algorithm.

Chapter 4: ImmunoGReP on APPSOMA: A cloud based ImmunoGenetic Repertoire analysis Pipeline for next generation sequencing data

INTRODUCTION

The vast amount of information gained from deep sequencing of the immunoglobulin repertoire offers a newfound ability to detect genetic disorders in B cells, quantify the efficacy of an immune response, and discover antigen specific antibodies. As a result, this promising technology for high resolution analysis of the humoral immune response is becoming widespread throughout immunology. However, the potential of next generation sequencing is exceedingly reliant upon computational tools that can highlight biologically relevant features against a backdrop of millions of sequence transcripts. Our dependency on computation will only intensify as the throughput of NextGen sequencing continues to increase. The requirement for robust computational tools is further burdened in repertoire analysis because of the discrepancy between immunologists with limited computational backgrounds and bioinformaticians with limited knowledge of immunologically relevant analyses[133]. There has been significant effort in immunology to close this gap between technical skills, and produce established methods for the analysis of antibody repertoires[8], [134], [135]. For example, novel analytical methods are currently available for V(D)J assignments, CDR3 spectratyping, multiple sequence alignment and clustering of clonally related antibodies, statistical analyses of antibody diversity, and visualization of relevant biological features[67], [68], [113], [117], [136]–[138]. While novel analytical methods are constantly introduced to handle challenges in

informatics, their value to immunologists is limited by their ability to provide these algorithms in a simple, openly accessible, and reproducible manner[139]–[141].

Moreover, many algorithms are stand-alone programs that were written in diverse programming languages and only address a small fraction of analyses required for standard repertoire studies (Table 4.1). This limitation forces immunologists to develop in-house bioinformatics pipelines where multiple tools are combined together in an analysis chain. Few research labs have the capability and computational expertise required for designing such analytical pipelines which entail installing foreign operating systems and programming languages, understanding which algorithms are best suited for specific experiments, and benchmarking the accuracy of results[141], [142]. These isolated analytical pipelines will not be viable options as the field of repertoire sequencing continues to expand and there is a greater demand for meta-analyses of previously published studies. Instead, there must be a centralized source of easily accessible analytical tools directed towards immunological repertoire research [8, p. 2014], [143].

As with all life sciences heavily dependent on computational techniques, this collection of analytics must provide scientists, having limited computational experience, with 1) novel algorithms that yield reproducible results 2) access to previously published sequence data, 3) the ability to benchmark performance of various informatics, and 4) the ability to combine multiple tools together into an analysis chain. Most importantly, as our understanding of humoral immunity progresses, it is essential that any bioinformatics pipeline is easily adaptable to new techniques and methods. There have been many recent initiatives to provide the scientific community with platforms which allow for the development of such open-sourced, reproducible, and standardized software suites[139], [141], [142], [144], [145]. One novel platform, called APPSOMA, is a cloud-based source

for sharing collaborative research and the development of novel computational methods[144]. In addition, APPSOMA allows bioinformaticians to deploy novel algorithms as "APPs" which allow non-computational users the ability to run these programs from any computer via a simple user interface.

We have used APPSOMA to design an ImmunoGenetic Repertoire analysis Pipeline (ImmunoGReP). ImmunoGReP provides access to algorithms for standard methods in bioinformatic analyses of antibody repertoires. Specifically, we offer programs for CDR3 spectratyping, IgBlast immunoglobulin analysis, and statistical methods for the comparison of repertoires. Most importantly, these programs are designed to run completely within the APPSOMA environment, and without the hassle of software installation or working in the often cumbersome Linux environment. Finally, in an effort to address problems in replication, transparency and curation of repertoire studies, we have also developed a database designed specifically for the storage of immunological sequence data. Access to this database is also provided through interfaces designed in APPSOMA.

Analysis		Available Programs	Language	Notes
Annotation of sequence data (V(D)J Assignments)		IMGT[67]	Closed-source	Web based only
		IgBlast[68]	C++	BLAST based
		iHMMune Align[113]	Java	Hidden Markov Model
		SoDA/SoDA2[146]	Web based	T-cell V(D)J Assignment
	Data analysis	Usearch clustering[19]	c++	Multiple alignment of DNA sequences
		Ig-HTS-Cleaner[147]	JAVA	454 sequencing error correction
		ClonalRelate[148]	JAVA	Agglomerative Hierarchical clustering
		IgTree[149]	Javascript	Phylogenetic analysis
Data storage		N/A		Lack of accessible databases for HTS of antibody repertoires

Table 4.1: Summary of programs commonly used for repertoire analysis

Repertoire analysis is comprised of three principal stages: Data annotation of sequence data into its relevant V(D)J gene segments and identification of the CDR 1, 2, and 3; Analysis of sequence data (i.e. multiple sequence alignment, and statistical analyses); Methods for storing sequence data and analysis. The above table illustrates some methods for performing these tasks. These programs are written in a variety of programming languages, perform only one task, and may require some computational background. Most importantly, there is no tool offered for the storage of antibody sequence data.

METHODS AND DESIGN

Hardware and APPSOMA installation

APPSOMA must first be installed on computers which will serve as "executive nodes" for running programs. Data generated from programs will be stored temporarily on

the executive nodes. The name of the temporary storage folder is called the "scratch" folder. Every APPSOMA user who has access the executive node will have their own unique scratch folder, and be able to store their data independently from other users. For permanent storage solutions, APPSOMA provides functions for downloading and uploading information to warehouses. These warehouses must also be set-up and installed by the administrators of the executive nodes.

APPSOMA was installed on two executive nodes assigned to the University of Texas, termed UTexas1 and UTexas2. Each node has approximately 200 GB memory and 650 GB hard disk space. All University of Texas employees have access to these nodes. APPSOMA also offers open access to three more nodes: node-0, node-1, and node-2; however, these nodes have very little RAM and available hard disk space, and, thus, are not suitable for running most scripts.

Precompiled software packages required for ImmunoGReP

The python module, pymongo, is required to run scripts that depend on functions which access the immunological database. For example, any script that queries sequences from an experiment would use the pymongo module. In order to use these functions, an administrator must install pymongo onto the executive nodes that will host and run these scripts specific to the database. However, this module is not required for other programs we offer which are not dependent on the database such as IgBlast analysis.

All other programs we offer do not need to be installed onto an executive node by an administrator. Instead, we provide scripts that will automatically install any necessary programs before running an ImmunoGReP APP. For example running the IgBlast APP will automatically download and install files and programs to the executive node that are necessary for running IgBlast. The following is a list of software tools that are

automatically installed onto the executive node when running scripts offered by ImmunoGReP: IgBlast[68], Circos[150], flash[151], and iCommands.

File handling and file formats in ImmunoGReP

The pathname and file location of any file created within an ImmunoGReP script follows a standardized set of rules. This standardization ensures that the user interfaces we design for running analysis APPs can easily locate files created within ImmunoGReP and exclude any file created by other APPSOMA programs. The following illustrates how files are organized in ImmunoGReP:

```
UserData/
    scratch/
        immunogrep/
            Exp_00001_Date_UserDefined
            Exp_00002_Date_UserDefined/
                    FastaFile.fna
                    FastaFile.imgt.annotation
                    FastaFile.imgt.query
                    FastaCorrelation.png
```

Specifically, all files are stored in a subfolder, called immunogrep, which exists within a user's main folder called the scratch folder (location in APPSOMA: ~/scratch/immunogrep). For every new analysis, a new experiment folder within immunogrep is created. This experiment folder will be automatically named by the current date and number of total experiments folders present in immunogrep; the last part of the folder name can be defined by the user. All files generated for a specific analysis, including figures, will be stored within this experiment folder. Finally, we also require that intermediate files created within an analysis pipeline have standardized extension

filenames. Result files created using immunoglobulin annotation programs such as CDR3 Spectratyping or IgBlast V(D)J assignment are always given a *.annotation extension file name. In addition, sequences that were queried from the database are saved to a file with a *.query extension filename.

Scripts within ImmunoGReP can handle the following file formats: fastq raw data files containing sequence quality scores returned by NextGen sequencers and (*.fastq), standard fasta file (*.fasta or *.fna), standard text-tab-delimited file (*.txt), and files written in JavaScript Object Notation[152] (*.JSON file). We have written functions which allow users to inter-convert their files between these four types. The JSON file format serves as the intermediate file format in all of our analysis scripts. That is, unless specified by the user, all scripts will by default output results file using the JSON file format. JSON file notation makes it simple to analyze multiple files that do not have the same structure. For example, in a text-tab-delimited file, each column name refers to a specific field in the file. That is, in a sample file, column 1 may refer to the CDR3 length, column 2 may refer to the CDR3, and column 3 may refer to the CDR3 frequency. Any script which requires this file must know that the CDR3 and its frequency are found in columns 2 and 3, respectively. However, this would not be a restriction if using a JSON file format. JSON files have no columns or column names; instead they use strings to refer to field names. The JSON structure for this file would be: {”Length”:6”,”CDR3”:”CARAEW”,”Frequency”:0.01} (Figure D.1). Because strings are used instead of standard column names, fields can be organized in any order. Therefore, we found a JSON file format to be the most amenable format to making an analysis pipeline from multiple scripts.

Bioinformatics analysis scripts offered by ImmunoGReP

ImmunoGReP is a compilation of scripts and programs written in tandem with the following users: Kam Hon Hoi, Benjamin Goetz, Andrew Horton, and Sebastian Schaetzle. The following sections below describe bioinformatics APPs that we currently offer for sequence annotation. In addition, appendix D contains a list of all scripts and APPs we have written for running ImmunoGReP. Scripts are openly available from the APPSOMA website (https://appsoma.austin.utexas.edu/ or https://appsoma.com/home). Finally, we have designed APPs, written in HTML, that provide a simple user interface for the following functions: 1) CDR3 spectratyping, 2) IgBlast analysis, 3) Comparing sequences across multiple datasets, and 4) querying the database for a list of sequences from an experiment.

CDR3 spectratyping

The CDR3 Spectratyping algorithm is based on a program written by a former member of our lab, Xin Ge (unpublished data). It has been modified, rewritten in python, and adapted for use in APPSOMA by Kam Hon Hoi. The CDR3 of an antibody is identified using conserved motifs in the framework 3 (FR3) and framework 4 (FR4) flanking regions of the CDR3. The motifs were identified using multiple sequence alignment of immunoglobulin sequences collected from the Kabat database, and followed by analysis of the conserved regions in the alignment that are directly upstream (FR3 motif) and downstream (FR4 motif) of the CDR3. For CDR3 spectratyping, both flanking motifs are converted into a probability weight matrix, V (20xN where N represents the length of the motif). Each column of the matrix corresponds to a specific amino acid position in the motif; each row corresponds to an amino acid. The values of the matrix, V(row,column), correspond to the probability that a specific amino acid (row) will appear in a specific

position along the motif (column). Using this probability matrix a position specific sequence matching (PSSM) algorithm is used to identify the region along the sequence read that best aligns to the motif of interest. Because the motif is defined at the amino acid level, all six frames of the sequence are aligned to both motifs. The algorithm only identifies a CDR3 when 1) both motifs are found above a threshold score and 2) both motifs are found on the same frame.

Isotype identification

The constant region or isotype of a NGS sequence is determined by the identification of a small oligonucleotide sequence, or barcode, found within the constant region of each isotype gene segment. Table D.2 lists the barcodes used to identify each sequence isotype. This script can only detect the isotype of immunoglobulin sequences whose cDNA, prior to sequencing, was amplified using primers that annealed downstream of the barcode sequence. The algorithm performs a simple gapless alignment between the sequence read and each possible barcode. An isotype is assigned to a sequence only if it aligns to the bar code in a region that has more than 80% pairwise nucleotide identity with the barcode.

IgBlast analysis

IgBlast analysis on APPSOMA runs the stand-alone IgBlast (verison 1.3.0) executable that was downloaded from the NCBI website. When running IgBlast analysis for the first time, it will automatically download the "internal data" folder from NCBI to the user's scratch folder. This folder is essential for IgBlast to run. Next, IgBlast analysis will run the standalone program using parameters defined by the user in the IgBlast user interface APP.

Results from IgBlast analysis are output using the BLAST output format 7 parameter. A python script was written to parse the output file and identify the following immunological features:

1) V, D, and J assigned germline gene segments

2) The sequence chain type (heavy or light)

3) The positions of the framework regions 1-4, CDR1, and CDR2

4) The prediction of the V-D and D-J junctions (CDR3 position)

5) The Number of mutations, insertions, and deletions in the sequence

6) The sequence alignment between the sequence and the predicted germline

The user has the option to output the parsed results file in either a text-tab-delimited file format or a JSON file format. JSON serves as the default file format.

Database schema

The database was designed around the Mongo NoSQL database language. A NoSQL database was chosen for two reasons. First, a significant advantage of NoSQL databases is that they scale with size efficiently by "scaling-out". This scaling means that a NoSQL database can be distributed or "sharded" across many computers or "nodes". When a database reaches full capacity and runs out of storage space, then another computer, of similar performance, is simply appended to the database cluster as a new "node". A second appealing aspect of a NoSQL database is its adaptability in that the schema of the database can be easily modified. Therefore, because we expect that this database will rapidly expand in size due to NGS, and that the relevant information we want to store will fluctuate greatly, a NoSQL database was a promising design. In a NoSQL database, data is stored in individual files called "documents". "Documents" are grouped together into large "collections". For example, our sequences "collection" groups together all NGS

transcript “documents”. We store specific pieces of information for each transcript “document” such as “CDR3” and “assigned V_H gene”.

The immunological database consists of four separate collections (Table 4.2):

A) Species collection: The species collection stores information regarding every species that has been analyzed. For each species we store 1) the species name, 2) its respective genus name, and 3) the NCBI assigned numeric ontology for that species.

B) Germline collection (Table D.5): The germline collection was created using information downloaded from the IMGT database of germline gene segments (version 3.0) [130]. Each document within this collection refers to a specific gene locus within the germline (i.e all genes composing V_H gene segment of humans). For each gene locus we store 1) the species name, 2) the germline locus ($V_{[H,\kappa/\lambda]}$, D_H, or $J_{[H,\kappa/\lambda]}$), 3) the source of the gene segment information (i..e IMGT), and 4) a subdocument which contains all of the information regarding every allele within the gene locus. This subdocument for each germline gene stores: i) the gene name, ii) the allelic name, iii) and its corresponding nucleotide sequence.

C) Experiments collection (Table D.3): The experiments collection stores all experimental information regarding next generation sequencing experiments that were added into the database. Specifically, the collection stores all of the metadata for an experiment (Table D.1). The following fields are indexed to support querying the experiment collection: Project_ID_INDEX, Experiment_Id_Index, Unique_Experiment_Id, Sequencing_Platform, Pairing_Technique_Index,

Isotype_Index, Chain_Types_Sequenced, Cell_Type_Indexed, Write_Access_Index, Species, and Publications_Index.

D) Sequences collection (Table D.2): Each document in the sequence collection refers to a specific DNA amplicon that was sequenced via NGS. Figure 4.4 describes what information is stored for each amplicon. To support queries, the following fields have been indexed in the sequences collection: Mongo_Experiment_ID, Paired_ID, Write_Access, Read_Access, Productive, chain type, CDR1, CDR2, CDR3, V genes, D genes, and J genes. The Paired_ID field will be used linking documents within a collection that refer to a biological heavy-light chain antibody. Read and write access define which members in APPSOMA have permission to read and update information for that specific sequence or document. The Mongo_Experiment_ID links the sequence to an experiment defined in the experiment collection.

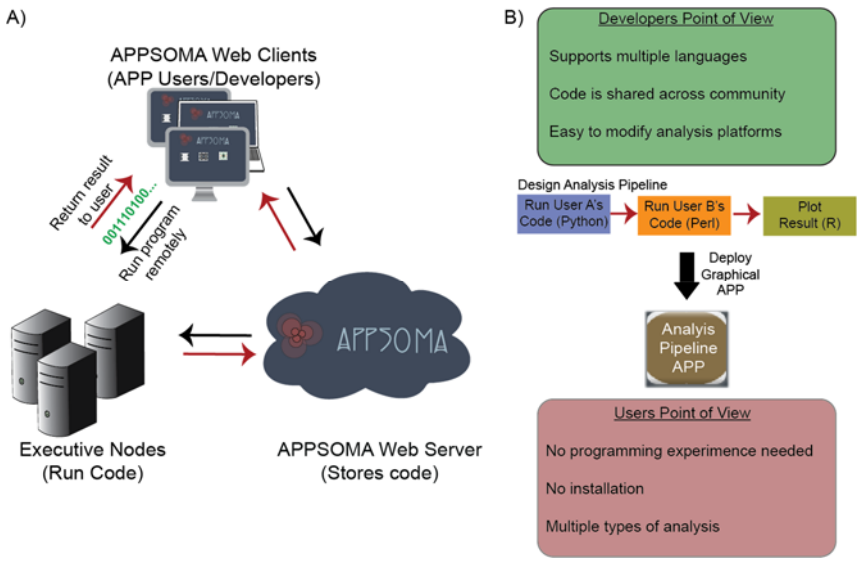


Figure 4.1: Description of APPSOMA

A) Users run algorithms on remote computers called "Executive Nodes" from a web browser B) Within APPSOMA, developers can program scripts in multiple languages and subsequently combine them into a bioinformatics pipeline. This pipeline can be made into an APP that has a graphical interface so that users, without computational backgrounds, can easily run the script.

One of the biggest challenges in designing novel algorithms for a scientific community is the balance between ease-of-use and ease-of-customization. That is, if one wants to create a tool that is highly customizable with many features then users will be required to have some computational background. In contrast, if one wants to make a program with a simple user interface that is easy for anyone to use, then the program must come precompiled and cannot be modified. APPSOMA[144] was created to alleviate such difficulties.

APPSOMA is a web-based platform where programmers can design source code from a remote computer and subsequently deploy their finalized programs as interactive

APPs. These programs and interactive APPs are run from remote computers referred to as "executive nodes" (Figure 1A). APPSOMA enables four intriguing properties for building a bioinformatics pipeline (Figure 1B): 1) centralized set of computation resources, 2) support for multiple programming languages, 3) collaborative hub of developers, 4) and APP deployment. First, because APPSOMA is installed on executive nodes, users can run these programs remotely from any computer with internet access. Second, APPSOMA is considered "language agnostic" in that it supports programs written in Python, Perl, R statistical programming, Bash, Ruby, and javascript. This support for languages commonly employed in bioinformatics, makes it easy to, not only import well validated programs, but also build analytical pipelines of programs written in a variety of languages. Third, APPSOMA allows easy collaboration between scientists. Users can be granted access to read, modify, and or run code written by fellow collaborators. Finally, the ability to convert programs into interactive APPs allows easy and reproducible access to published algorithms. In summary, these features provided an ideal platform for creating a centralized and open-sourced immunological analysis platform.

RUNNING IMMUNOGREP ON APPSOMA

The analysis of immunoglobulin repertoires often involves three principal steps: 1) annotation of sequencing data such as V(D)J assignment, 2) analysis and comparison of NGS experiments, 3) and the storage of sequencing data and results. The following discuss how ImmunoGReP was designed to provide solutions for each feature.

Solutions for analysis of NGS immunological data

One of the most important features of ImmunoGReP is that the analysis pipeline is not dependent on just one analysis or annotation program. Instead, because different

algorithms serve different needs, we wanted to provide users with the ability to choose between algorithms. Most programs we offer are designed to be "modular" in that they perform highly specific tasks, but can also be easily chained together into functional bioinformatics pipelines. Therefore, this modular structure within APPSOMA, enables us to insert any new combinations of techniques within an analysis pipeline regardless of upstream or downstream processes.

APPS for Immunoglobulin Annotation

First, we offer users the ability to analyze sequencing data using the well-established and validated IgBlast algorithm for annotating B and T cell receptors sequenced from human, mouse, rabbit, and sheep repertoires[68]. IgBlast analysis will align each immunological sequence read to its predicted V, D, and J germline gene segments. In addition, IgBlast predicts the framework regions, the CDR regions, and the number of nucleotide mutations (SHM) present when aligned to its assigned germline gene.

Running the stand-alone IgBlast program on a local computer is not exceptionally easy for non-computationalists. Researchers must first make sure they have installed a Linux operating system that contains the proper environment settings and all the necessary packages for running the IgBlast source code. Another source of difficulty is that the output file of an IgBlast analysis is written in the BLAST output format. Because this format does not store sequence information using delimiter-separated values (i.e. tab delimited file), the results cannot be presented in a tabular format using spreadsheet programs such as Excel. This file format can be problematic for downstream analysis because most scripts rely on using simple delimiter values to separate the annotated features (i.e. CDR1 and CDR2) from each sequence read into individual columns for impending analysis. Thus, after running IgBlast, additional steps are often required to parse the BLAST formatted output

file, identify the features of interest to the researcher, and output the result in recognizable formats such as tab-delimited files.

The IgBlast stand-alone program has been repackaged into scripts hosted on APPSOMA. These scripts address all of the extraneous processes associated with the stand-alone IgBlast analysis. Users can run IgBlast from a graphical user interface that works within the ImmunoGReP pipeline (Figure 4.2). When users run the APP, IgBlast will be automatically installed on the executive node running the program, then run IgBlast with the necessary settings, and finally report the IgBlast results in an appropriate format defined by the user.

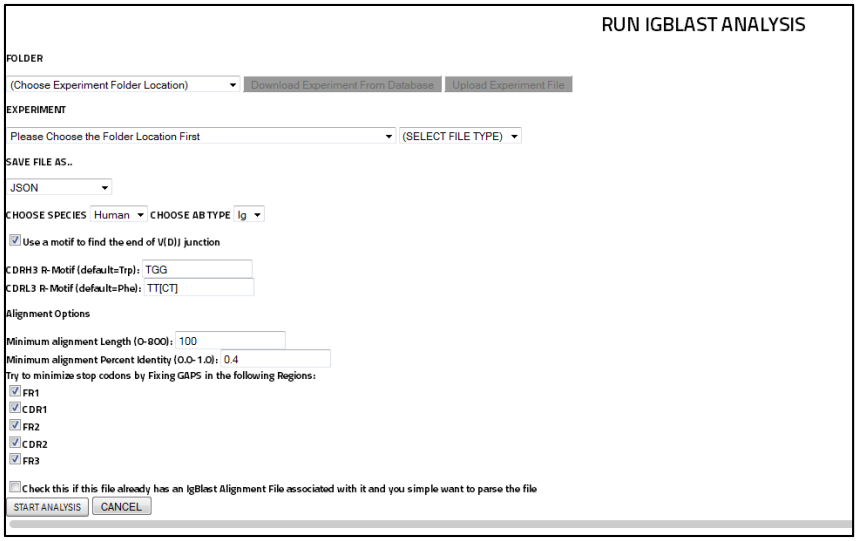


Figure 4.2: Example of simple user interface for running IgBlast in APPSOMA

Although IgBlast is a highly useful and powerful tool, it is not designed for all types of repertoire analysis studies. First, IgBlast analysis cannot always handle, in a reasonable time, the quantity of sequence information associated with contemporary repertoire studies. More importantly, IgBlast does not accurately predict the V-D and D-J junctions that form

the CDR3 ends of a recombined immunoglobulin sequence. This poses a problem for researchers that rely on CDR3 analysis rather than germline assignment. For example, CDR3 sequence spectratyping has been important for both clustering sequences into unique clonotypes and for identifying antibodies at the proteomic level via mass spectrometry[9], [10].

Therefore, in addition to IgBlast analysis, we also provide a program for extracting only the CDR3 sequence from sequencing data. This script, based on a previously designed in-house algorithm, relies on the identification of CDR3 flanking motifs found in the framework 3 and 4 regions of an antibody. Importantly, the motif we have selected is common to all germline V and J gene segments within a gene locus. Therefore, the CDR3 can be identified regardless of its respective V and J germline gene segments. Similarly to IgBlast on APPSOMA, we have designed this program with a highly customizable user interface for the simple analysis of NGS data. The CDR3 spectratyping algorithm is currently designed for the identification of the heavy and light chain CDR3s from multiple species including human and mouse. Additionally, because users are allowed to upload a custom motif table, the APP can be easily extended to the analysis of even more species, or other regions of the antibody, such as the CDR1.

The programs described above annotate the variable heavy and light chains of an antibody. In an effort to provide a complete and comprehensive analysis of sequence data, we also provide a program for determining the isotype of a sequence. The program aligns sequences to small barcodes which correspond to the constant regions of an IgG, IgM, IgA, IgK, or IgL antibody isotype. The correct isotype is assigned to a sequence containing more than 80% nucleotide sequence identity with the barcode.

APPs for repertoire analysis and visualization

Prepackaged programs such as IMGT and IgBlast only annotate sequence information. However, researchers often struggle the most with post-annotation analysis because they do not know how to analyze their data or run the relevant statistical tests. Therefore, we include programs which accept output files from either IgBlast, IMGT, or CDR3 spectratyping annotation and generate "descriptive statistics" of the repertoire of interest. These statistics, such as CDR3 length distribution, V and J gene segment usage, and CDR3 diversity, are common methods for characterizing the adaptive immune response. In addition, we designed an APP which will compare the overlap between sequences shared across multiple experiments and subsequently calculate the pairwise correlation between datasets. These programs will also generate plots, such as histograms, heat maps, and circus plots, which summarize the analyses (Figure 4.3)[137]. It is important to stress that because all analysis APPs are provided within the APPSOMA environment, the user can both annotate and analyze sequence data in one entire step. Without such a pipeline, a researcher without any computational background would be forced to first annotate sequence data using IMGT or IgBlast, export the results, convert the file into a file format that is compatible with a downstream analysis program, run the

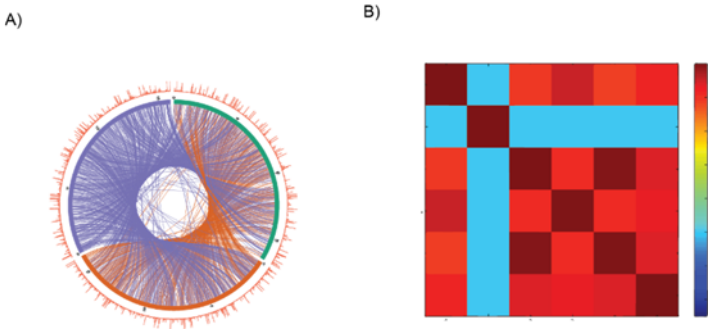


Figure 4.3: Examples of output from analysis APPs

A) Circos plot of population overlap B) Heatmap of pearson correlations

second analysis on the annotated sequences, and finally plot the results of the analysis using a third program.

Solutions for data storage

Systems for data curation and storage are equally as important as systems for analyzing data. We aimed to create solutions which address proper curation of experimental data, long term storage of raw data, and storage of annotated NGS immunology data (Table 4.2).

Database Collection	Description of collection	Total # of documents	Description of document	Species (# documents)
IRODS	Data curation; Storage for raw data; IMGT annotation	155	All files associated with an experiment but not processed in APPSOMA (fastq and IMGT)	Mus musculus (79); Homo sapiens (64); Capra aegagrus (3); Gallus (1); Oryctolagus cuniculus (6); Ovis aries (2)
Experiments Database Collection	Data curation; NGS Immunoglobulin experiment information	21	A specific NGS experiment	Mus musculus (7); homo sapiens (10); Capra aegagrus (2); Ovis aries (2)
Sequences Database Collection	NGS Immunoglobulin sequence information	30,279,604	A NGS immunoglobulin sequence that has been annotated (i.e. IMGT/IgBlast)	Mus musculus (153480); homo sapiens (30030988); Capra aegagrus (34454); Ovis aries (60682)

Table 4.2: Summary of information currently stored in NGS immunological database

Standardization of experimental info for raw data

Data processing and analysis is dependent upon the experimental methods used before sequencing. Factors such as the number of immunoglobulin cells analyzed, the

cellular markers used to isolate specific cell subsets, and the inclusion of barcodes or replicate samples affect how sequencing data should be analyzed and interpreted downstream. We wanted a system that stored both the raw sequencing data for an NGS experiment and information describing the experimental process. The online data management tool termed integrated Rule Oriented Data System, or iRODS, is an ideal tool for such a system[153].

iRODS emphasizes secure backup solutions for research groups and easy distribution of data across collaborators. Moreover, files uploaded to iRODS can be tagged with an unlimited number of fields, called metadata, which can be subsequently used in queries for specific experiments. For every NGS repertoire study, we stored the following files in iRODS: 1) all raw files generated from NGS, 2) quality filtered FASTA files, 3) and any sequence annotation results obtained from analyses performed outside of APPSOMA such as IMGT analysis of immunoglobulin sequences. We also enforced descriptive fields that must be included when uploading a sequencing dataset. These fields include, but are not limited to, 1) the project name and description, 2) number of cells analyzed, 3) relevant cell markers, 4) sequencing platform used, 5) and lists of members who have read and write access to the experimental data (Table D.1). Regardless of any changes we make to the ImmunoGReP database or analysis tools, iRODs presents a safe solution for permanent backup of all raw sequencing data and experimental information.

An immunological HTS database

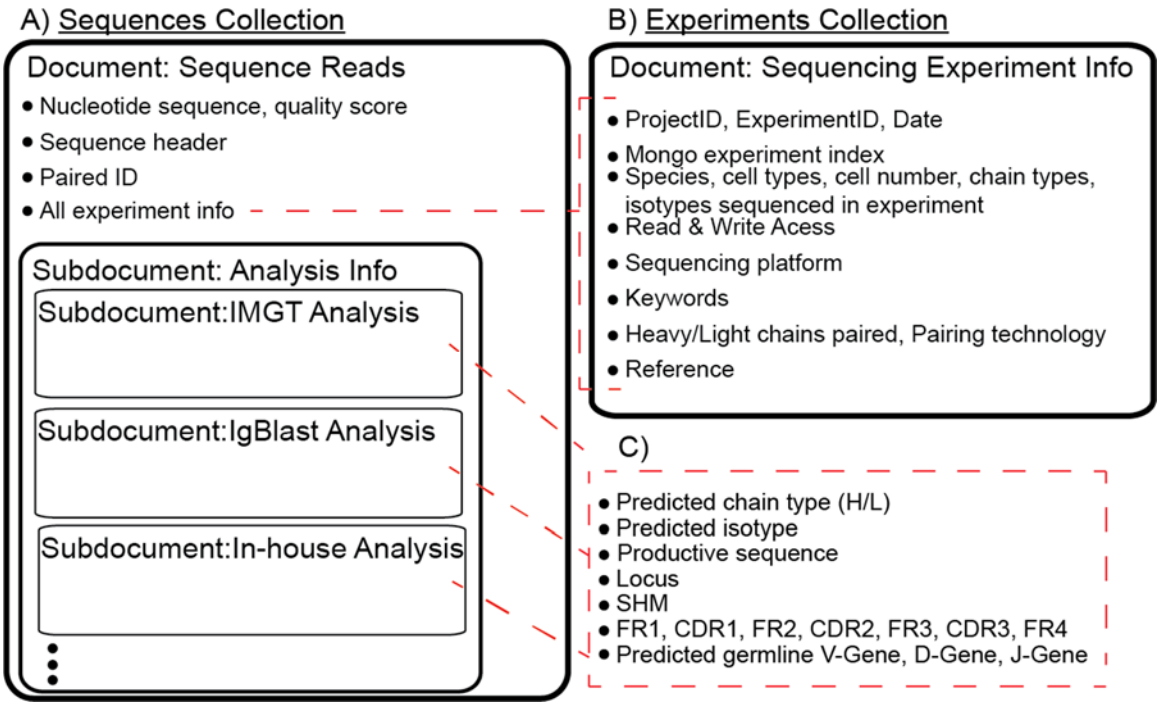


Figure 4.4: Simplified database schema for the sequences and experiments collections

A) Each NGS sequence read is stored as an individual document in sequences collection. Each document contains information about the sequence and the experiment it belongs to B) Condensed experiment collection that only contains information on each repertoire study. C) For every sequence read, analysis information is stored within the sequences collection. Analysis from different algorithms such as IMGT and IgBlast are stored in separate subdocuments

The iRODS system discussed above cannot store detailed information provided by annotation tools such as IMGT and IgBlast. Instead, it is necessary to organize analyzed data into a defined database openly available to the community. For example, a future meta-analysis study may search for previous experiments which contained immunoglobulin sequences with a specific CDR3 amino acid sequence. We aimed to create a database that could facilitate such prospective endeavors.

Figure 4.4 illustrates a simplified schema of how our database is currently organized. In summary, the database is organized across two collections. The first collection is called the experiments collection and stores the experimental information stored in iRODS metadata. Transcript specific information from all experiments is stored in a very large collection called the sequences collection. The sequences collection stores information regarding biologically relevant features determined from sequence annotation programs such as IMGT. Such features include the predicted V, D, and J genes, CDR 1, 2, and 3 sequences, framework regions, and productivity of an immunoglobulin sequence.

However, different annotation programs will report varied results for the same features listed above. For example, both IMGT and IgBlast use separate nomenclature for the set of mouse germline V_H genes. Because of this lack of standardization, the database must accurately document which programs were used to analyze sequence data. Consequently, for every stored sequence read, we will store annotation generated by different algorithms separately within the analysis field (Subdocuments in figure 4.4A). That is, sequences analyzed using both IMGT and IgBlast, will contain two different analyses that are stored separately within the document. Using this method, researchers can easily compare results reported by IMGT and results reported by IgBlast. Currently, we store information compiled from running IMGT, IgBlast, and our in house CDR3 spectratyping pipeline.

All functions for inserting experiments into the database have been implemented solely in the APPSOMA platform. Table 4.3 summarizes the performance of our database solution and sample queries. We currently have 30 million sequences stored in the database compiled from 21 experiments. In total, the size of this database requires 100 GB which accounts for space required for indexing all fields used for querying. On a standard

computer it takes 34 minutes to populate the database with 1 million sequence documents and 30 seconds to query the sequences from the database. A query for a CDR3 sequence found three times among these 10 million sequences takes under five minutes.

Database Performance	
Time to insert in database	34 minutes
Time to query from database	0.5 minutes
Size of inserted documents	3 GB (~3 KB/document)

Table 4.3: Performance of Database Functions

This table summarizes the time it takes to both insert an analysis file containing 1 million sequences and subsequently query those documents from the database. It also shows the average amount of disk space each document takes.

The combined pipeline for repertoire analysis: annotation, analysis, storage, and validation

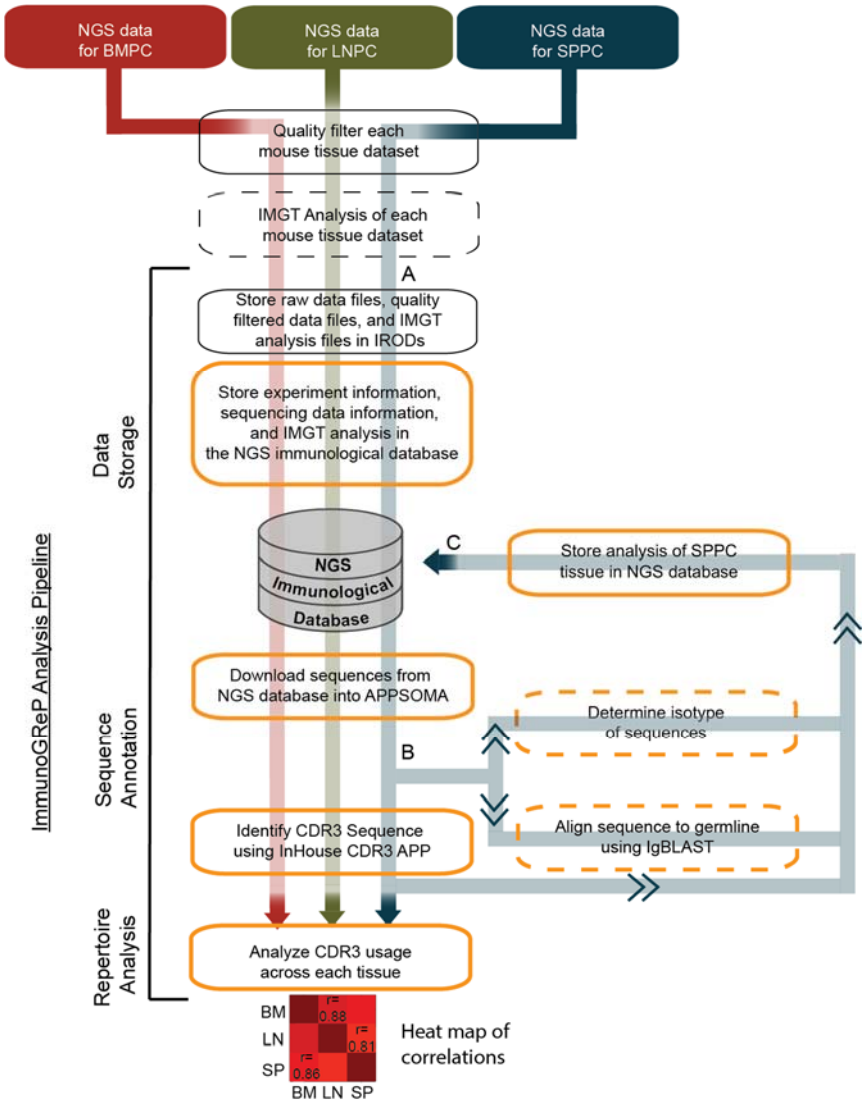


Figure 4.5: Example of repertoire analysis using ImmunoGReP on APPSOMA

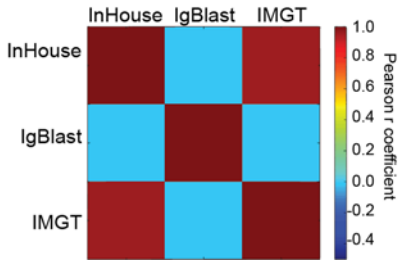
Example analysis pipeline for the lymphoid tissue repertoires (BMPC, LNPC, and SPPC) from the mouse 23 NGS datasets described in Chapter 2. Dotted outlines represent optional steps; Yellow boxes represent APPs. All data is backed up securely in IRODs (A). APPs report all relevant information for repertoire sequence data (B). Analysis from APPSOMA can be stored in the NGS immunological database (C).

Chapter 2 describes the comparison of lymphoid tissue repertoires isolated from mice immunized with the HEL antigen. The analysis for that study was performed using IMGT annotation. Figure 4.5 illustrates how, using all analyses discussed above, the ImmunoGReP pipeline can reproduce results from that experiment in less than one hour. It should be stressed that this pipeline includes programs written in multiple languages such as python, pymongo, c++, and perl. Users who choose to run the programs from within APPSOMA will not need to worry about running analyses using multiple computer languages. Instead, all steps performed within APPSOMA, illustrated by yellow outlines in Figure 4.5, can be performed without any computational experience. Finally, this modular pipeline can be easily modified.

We next demonstrate that the ImmunoGReP pipeline can be used to benchmark various algorithms. Centralization of both immunological data and analysis, enables researchers to easily compare multiple algorithms using identical sequence sets and computer hardware. As an example, we can use the LNPC dataset stored in the database to compare CDR3 identification using our in-house script or IgBlast to CDR3 sequences reported by IMGT. Using the pipeline, it immediately becomes apparent that IgBlast fails when trying to report the heavy chain CDR3 sequence of variable genes (Figure 4.6A). Our InHouse CDR3 script on the other hand, shows very strong similarity with respect to IMGT (Pearson correlation ~ 0.95) generated results. Further inspection of the top CDR3 sequences in each analysis reveals that although IgBlast identifies the start of the CDR3 correctly, it does not provide the same DJ junction reported by IMGT (Figure 4.6B). This example demonstrates that in order to use IgBlast to provide accurate CDR3 spectratyping, an additional script would be required to accurately identify the DJ terminal junction.

Thus, we show that ImmunoGReP provides a centralized source for researchers to evaluate algorithms and select those that best meet their research goals.

A)



B)

CDR3 Identification Method	Sequence	Counts
InHouse	AREYGGRGFDY	1103
IgBlast	AREYGGRGF - -	1244
IMGT	AREYGGRGFDY	983

Figure 4.6: Comparison of CDR3 identification across algorithms

A)Heatmap of pearson correlations (generated in ImmunoGReP) comparing methods for identification of the CDR3 amino acid sequence using the same test set B) Top CDR3 identified in each method

DISCUSSION

We have presented the framework of a cloud based tool for analyzing immunoglobulin repertoires. ImmunoGReP is one of the first "suite of programs" designed to address all aspects of a repertoire study. Specifically, we present methods for data curation, sequence annotation, data analysis, data visualization, and data storage. This outlined pipeline represents a first stage for ImmunoGReP and demonstrates the utility of a centralized source of analysis tools. The pipeline provides multiple solutions for repertoire analysis and can be easily modified to accommodate novel algorithms or "modules". For the second stage, we want to continue to add to this suite of available programs for repertoire study. For example, we plan to provide accessibility to algorithms for clustering, in-silico pairing of heavy and light chains, and additional methods for sequence annotation. In addition, because the database of germline gene segments is always changing, the IgBlast germline database may not always provide the most updated

or comprehensive set of germline gene segments for V(D)J assignment. Therefore, the next version of the APPSOMA based IgBlast APP will also permit the alignment of NextGen sequence data against a custom database of germline gene segments defined by the user.

Providing access to a database enables open accessibility and reproducibility of repertoire sequencing data. While we currently only allow users to download sequencing data from a previous study, we plan to expand on this functionality and enable advanced queries for biologically relevant features (i.e. sequences expressing specific CDR3 amino acid sequences). Finally, a centralized source of analyzes tools connected to a database enables researchers to easily benchmark novel algorithms and gauge their effectiveness. In lieu of this potential, we also plan to create a gold test set, compiled from repertoire data of multiple species, that can used for benchmarking algorithms. In summary, using the APPSOMA platform, we present simple access to a pipeline of algorithms which provide a standardized, transparent, and reproducible analysis of immunoglobulin repertoires.

Chapter 5: Conclusion and Future Aims

METHODS FOR IMMUNOGLOBULIN REPERTOIRE ANALYSIS

The humoral adaptive immune system is composed of an impressively diverse repertoire of antibodies that display varied specificities against unforeseen pathogens. This vast repertoire is attributable to the evolution of an intricate system of genetic diversification processes which generate novel defensive elements during the lifetime of an individual. High throughput analysis of the immunological repertoire via next generation sequencing has the potential to deconvolute many aspects of adaptive immunity. Such advancements can have a profound effect in vaccine development, treatment of autoimmune disorders, and antibody discovery.

Nevertheless, this technology is still in its infancy and, as a result, there are many challenges that must be overcome. One of the largest challenges is that the computational and statistical methods needed for the analysis of sequenced repertoires are unfamiliar to many researchers in immunology. In this work we introduce many informatics tools for the statistical analysis and visualization of large repertoire datasets. Additionally, we introduce techniques and algorithms that can efficiently process large amounts of data.

In this work we show that the challenges associated with systems analysis of immunological repertoires can be addressed by analogous problems in unrelated fields. Identification of such analogous situations could be highly beneficial to the progression of immunology research. One example application is the extension of Fast Fourier Transform techniques for V(D)J annotation of immunoglobulin sequences. Specifically, we show a novel use for sparse Fourier transforms in identifying the best regions of alignment between an immunoglobulin sequence and its germline gene. Moreover, statistical methods required for the analysis of large datasets and comparisons of trends between populations

can be also be found in analogous fields. Statistical methods used in ecology can provide a framework for immunoglobulin sequence analysis. For example, the diversity of repertoires can be calculated and compared using numerous indices of diversity previously investigated by ecologists.

In addition, progress in high throughput sequencing of immunological repertoires must address how experimental error affects quantification and characterization of an immune response. Our analysis of lymphoid tissues demonstrated that the immunological repertoire is highly variable across individuals, and, in addition, we could only identify one mouse that presented a strong immune response against the antigen. While we were able to identify statistically robust trends in the global characteristics of lymphoid tissue repertoires, it was difficult to ascertain the significance of very minute, yet potentially biologically important, differences within the repertoire tissues. Because there are not many quantitative characterizations of potential experimental error and replication, we repeatedly exemplified the use of non-parametric statistical analyses in our study of the lymphoid tissues. It is important for future studies to give substantial focus to designing controlled experiments which allow for robust statistical analyses of significance and demonstrate the effects of experimental bias and sequencing error on interpreted results.

INVESTIGATING METHODS FOR ACCURATE DEEP SEQUENCING OF THE CDR3

The majority of methods described in this work concern the downstream analysis of repertoire studies after high throughput sequencing. However, there are still significant areas of improvement in the upstream design of cDNA libraries that would greatly enhance the accuracy of sequencing. This need arises because the majority of repertoire studies entail the identification of unique mutations that correspond to specific sequences of the BCR. Such high resolution sequence analysis is difficult in the background of NextGen sequencers that have an intrinsic average error rate of 10^{-2}-10^{-4} mutations per base pair.

Therefore, a future aim is the design of experimental methods which can mitigate

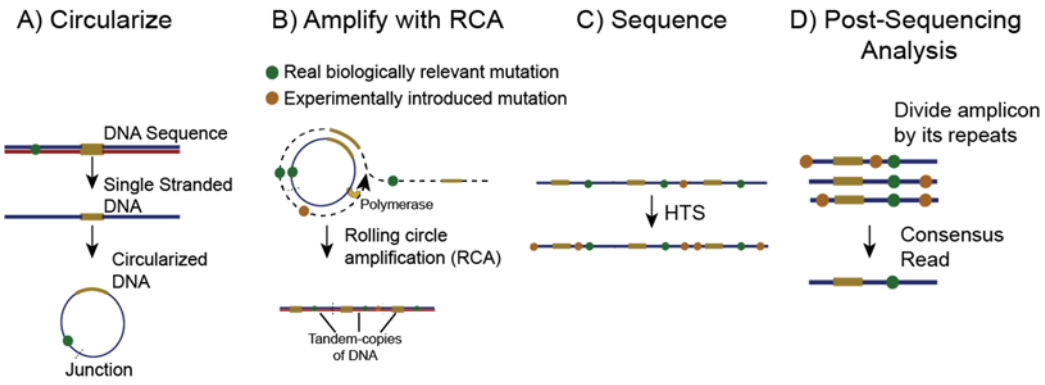


Figure 5.1: Circle Sequencing Protocol

> Circle sequencing is a method for correcting error introduced during NGS. A) Circularize library of single stranded DNA B) Perform rolling circle amplification (RCA). This amplification step synthesizes tandem copies of circular molecules. C) Perform NGS on the amplicon with tandem repeats. Non-biological errors are introduced randomly (orange circles) during HTS D) From the sequenced read, identify the repeats generated by RCA. Align each repeat from the sequence read and make a consensus sequence, eliminating error introduced on individual copies.

non-biological sequencing error. Our approach will take advantage of a novel method for library preparation, termed "circle sequencing", which corrects sequencing errors computationally [154]. In circle sequencing, DNA amplicons from the cDNA library of

the immune repertoire are circularized and subsequently replicated using a rolling-circle polymerase which creates tandem copies of the sequence that are physically linked on a double stranded DNA product (Figure 5.1). Because each tandem copy is independent, it is unlikely to observe the same type of sequencing error introduced across multiple copies. Therefore, these tandem copies can be used, post-sequencing, to form a consensus sequence and correct for any errors that were introduced by next generation sequencing. The application of circle sequencing for sequencing the yeast genome resulted in a decrease in error rate by three to four orders of magnitude [154].

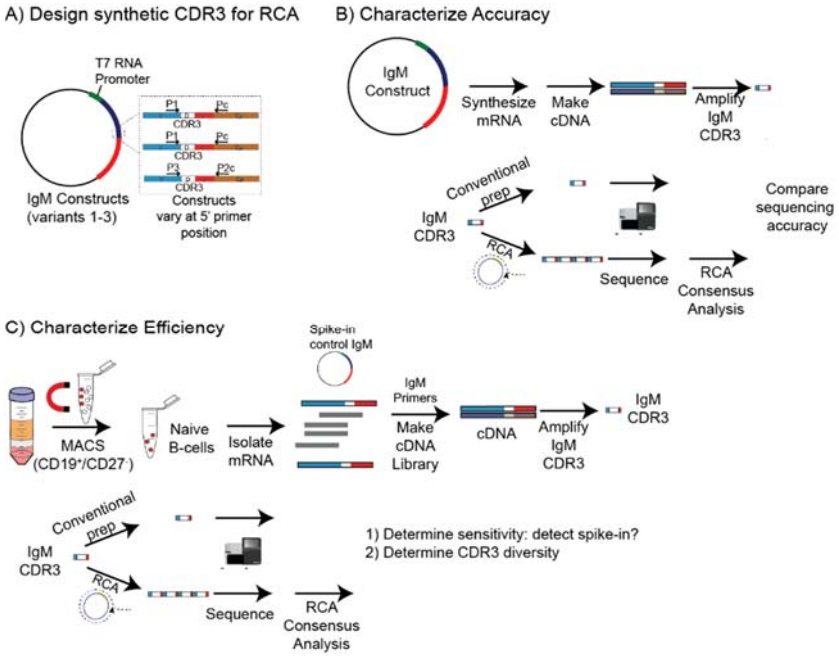


Figure 5.2: Testing RCA as a method for CDR3 spectratyping

A)Variant that transcribes control IgM construct from T7 B) Analyze error rate of NGS of control IgM using RCA C) Isolate human naïve B cells and extract mRNA. Spike in IgM control, make cDNA, and sequence cDNA library. Test whether RCA can detect spiked in control IgM

We will investigate whether circle sequencing can be an accurate method for deep sequencing of the CDR3 region, the most diverse region of an antibody sequence. Figure 5.2 illustrates the experimental protocol for evaluating circle sequencing. We want to characterize both the accuracy of CDR3 sequencing and the sequencing efficiency when using RCA as a method for repertoire analysis. Sequencing efficiency will refer to the fraction of lymphocytes whose CDR3 can be accurately detected and sequenced using RCA. To address both questions, we have made a synthetic construct of a full length IgM immunoglobulin whose transcription is regulated by the T7 RNA polymerase promoter (Figure 5.2a). NGS sequencing of the IgM mRNA generated from this construct using both conventional and RCA sequencing will demonstrate the accuracy of the RCA method (Figure 5.2b).

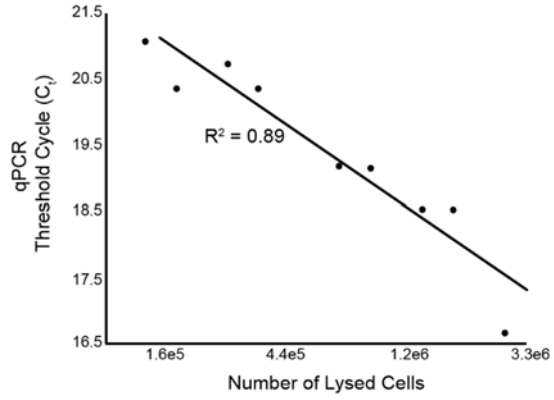


Figure 5.3: Quantification of IgM transcript in Naïve B cells

Scatter plot shows the results from qPCR analysis of the levels of IgM transcript in naïve B cells. Naïve B cells were isolated from concentrated leukoctyes using MACS isolation kit for CD19+ and $CD27^-$ markers expressed on naïve cells. The concentration of isolated naïve cells was measured using a countess automated cell counter. mRNA was extracted from samples containing a varied number of isolated naïve B cells (3e6 to 1.5e5). mRNA was converted to cDNA and qPCR was subsequently performed on the cDNA samples using primers specific for IgM constant region.

Second, to demonstrate sequencing efficiency, we will spike-in known amounts of IgM mRNA into mRNA isolated from naïve B cells. Subsequent sequencing of this library using both conventional and RCA sequencing will determine whether either methods is sensitive enough to detect the synthetic CDR3 sequence (Figure 5.2c). The amount of control mRNA IgM spiked-in will be determined using qPCR analysis which will quantify the average amount of IgM mRNA expressed by naïve B cells (Figure 5.3).

EXTENSION OF THE IMMUNOGREP ANALYSIS PIPELINE

As a final note, there is still a significant amount of work that is required to ensure standardized analysis methods in immunology. This work is dependent upon the efforts contributed by the immunology community as a whole. Consequently, we have introduced the ImmunoGReP analysis pipeline. ImmunoGReP was designed to facilitate immunoglobulin repertoire analysis by creating an open source of cloud-based tools. We also created a database designed specifically for immunology research. Some current areas of immunoglobulin analysis that we have not yet addressed include 1) the availability of novel algorithms for germline annotation such as our in house germline assignment program, 2) programs which take advantage of sequence quality scores to correct for immunoglobulin sequence error, 3) algorithms for clustering and phylogenetic analysis, and finally 4) the integration of an analysis pipeline for proteomic studies. While, our pipeline provides non-computationally skilled researchers with easy access to the basic tools for analyzing repertoires, it was designed to be continually expanded. As our knowledge of the adaptive immune system continues to develop and change, so will the compilation of analytical tools and algorithms. Access to a constantly updated centralized source of standardized tools is essential for growth in this field. Therefore, the most

important future prospect for this pipeline is to promote collaboration and integration of analysis tools from multiple research groups within the immunology community.

Additional work: Investigation of *E. coli* redox chemistry

PREFACE

In addition to the study of immunological repertoires, a significant amount of work was dedicated to the study of redox homeostasis in *E. coli*. Many biological processes are closely coupled to redox reactions which transfer electrons from an electron rich source to an electron sink compound. In this regard, *E. coli* have evolved to tightly regulate electron flow using an intricate organization of redox-sensitive proteins and cofactors. Among these molecules, the cofactor, Glutathione, is one of the most essential for redox homeostasis in *E. coli*. Glutathione (GSH) is a small tripeptide, γ-glutamyl-L-cysteinyl-glycine whose intracellular concentration in bacteria ranges between 0.1 and 10 mM[155]. As the most prominent thiol species, GSH acts as a "redox buffer" regulating whether oxidative reactions in the cytoplasm are thermodynamically favorable[156]. In addition to maintaining a reduced glutaredoxin pool, glutathione has been attributed with protection against reactive oxygen species (Figure A.1)[155], [157].

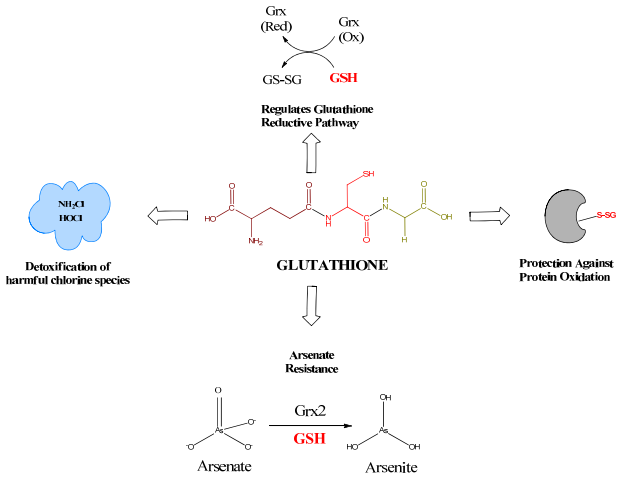


Figure A.1: Role of Glutathione in *E. coli*

Redox homeostasis is a highly plastic process across organisms and not every organism exhibits same dependency on Glutathione. In fact some organisms have evolved in the absence of Glutathione[158]. In such situations, other redox-sensitive proteins such as thioredoxins must address oxidative stress created by reactive oxygen species[159], [160]. Thus, we have been interested in novel methods in which *E. coli* would respond to selective pressure imposed from the removal of glutathione dependent pathways. Arsenate resistance is an example of a redox mechanism highly dependent on glutathione. E. *coli* mutants lacking glutathione, glutaredoxins, or arsenate reductase cannot grow in the presence of arsenate. Moreover, not all arsenate reductase enzymes are dependent upon glutathione[160]. Thus, the removal of either of these elements provides an effective selection pressure for screening novel electron pathways. The aim was to investigate whether these novel pathways could give insight with respect to the evolution of analogous arsenate reductase pathways. Interestingly, while we could not identify pathways independent of Glutathione, the results of our genetic screens did result in the identification of novel pathways for arsenate resistance. One of the most promising discoveries from our work was the identification of a Glutathione-S-transferase capable of catalyzing the reduction of arsenate in the absence of wild type arsenate reductase. The following study discusses the results of this novel finding. The inability to identify glutathione-independent pathways further illustrates that, while redox mechanisms may be a highly plastic process, it is very difficult to decouple Glutathione from redox homeostasis in *E. coli.*

AN ALTERNATE PATHWAY OF ARSENATE RESISTANCE IN *E.COLI*

Introduction

In nature, the toxic heavy metalloid, arsenic, exists predominately as oxyanions in either its pentavalent or trivalent oxidation states[160]. The natural abundance of both toxic molecules imposes a strong selection pressure for organisms to evolve arsenic detoxification pathways. Because these pathways have evolved in parallel across many organisms, the study of arsenic detoxification is often a topic of interest[160], [161]. Detoxification of reduced arsenite in-vivo can be achieved via sequestration or efflux from arsenite specific transporters[162]. Arsenate, on the other hand, is not immediately exported. Instead, the first step in arsenate detoxification across species has evolved to reduce arsenate into arsenite[160], [163], [164]. This reductive step is catalyzed by a diverse class of enzymes termed arsenate reductases.

Five mechanistically and phylogenetically distinct classes of arsenate reductase enzymes have been identified so far. While all families contain a catalytic redox-sensitive cysteine residue, the specific mechanism of action can differ significantly between classes [160]. Arsenate reductases from Gram-positive bacteria accept electrons from thioredoxins which are maintained in the reduced state by the NADPH dependent enzyme thioredoxin reductase[165]. These enzymes display structural similarity to mammalian tyrosine phosphatase enzymes and exhibit in-vitro phosphatase activity[166] [167]. Conversely, arsenate reductase enzymes from Gram-negative bacteria including *E. coli* depend on the glutathione/glutaredoxin reductive pathway [168], [169]. This class of reductases does not show evidence for phosphatase activity[170]. Given such a distinct subset of arsenate reductase enzymes, arsenate resistance evolution seems to be a highly

plastic process in which multiple unrelated enzymes can evolve to catalyze the essential reductive step.

In an earlier study we used genetic selection approaches to identify alternate electron transfer pathways for the reduction of As(V) in *E.coli*. Selection for suppressor mutations capable of rescuing arsenate sensitivity of a glutathione null *E. coli* mutant (Δ*gshA*) revealed instead a novel pathway for glutathione synthesis via the *proAB* operon, responsible for proline biosynthesis[171]. Suppressor mutations in *proAB* resulted in accumulation of γ-glutamyl phosphate which reacted spontaneously with L-Cysteine to form γ-glutamyl cysteine, the reaction product of GshA. Thus, the genetic screen resulted in the restoration of electron flux towards arsenate reduction[171]. These findings underscored the essentiality of the glutaredoxin/glutathione system in providing electron equivalents for As(V) reduction by ArsC.

We investigated whether other *E.coli* enzymes or pathways are able to confer resistance to lethal concentrations of arsenate in the absence of the arsenate reductase enzyme, ArsC. The complete set of *E. coli* K12 ORF ARCHIVE (ASKA library) can be readily used as a powerful screening tool for identifying new enzyme functionalities[172]–[174]. Using this genetic screen, we found that overexpression of the glutathione S-transferase B (*gstB*) gene from a multicopy plasmid enabled *E.coli* Δ*arsC* cells to grow in the presence of millimollar levels of arsenate. GstB was previously shown to be involved in bromoacetate resistance via a dehalogenation mechanism in which the conjugation of bromoacetate and glutathione results in the release of a free bromide[175]. In this study, we show that GstB also facilitates a glutathione-dependent arsenate reduction reaction. Whereas cysteine plays an essential role in the catalytic mechanism of the much better studied E.coli glutathione-S-transferase, Gst, we find that the two Cys residues in GstB,

which forming a thioredoxin-like CXXC fold, are dispensable for As(V) reduction[176]. Instead conserved arginine residues within the active site are critical for binding to electrophilic substrates including arsenate. In *vitro* and *in vivo* biochemical studies further revealed that GstB directly catalyzes the reduction of arsenate with GSH as the reductant in a reaction that does not depend on glutaredoxins. Thus, our results define an auxiliary, mechanism for arsenate resistance in bacteria.

Materials and methods

Reagents

Sodium Arsenate (S9663-50G), Sodium Arsenite (S225I-100G), and MOPS Salts (M1254-250G) were purchased from Sigma Aldrich. Disposable Strata-SAX anion-exchange columns (8B-S008-HCH) were purchased from Phenomenex. Isopropyl β–D-1-thiogalactopyranoside (206-703-0) was purchased from Fisher Scientific. All reagents for arsenic detection (EZ Arsenic Test Kit Catalog # 280000) including arsenic test strips, Zinc Sulfate, and Phosphate, were purchased from the Hach Chemicals. For the NADPH coupled assay, Glutathione reductase from Bakers Yeast (060M7405) and Reduced Nicotinamide adenine dincucleotide phosphate (N7505-25MG) were purchased from Sigma Aldrich.

Bacterial strains, plasmids, and media

Cells challenged for arsenate resistance were grown in MOPS minimal growth media containing minimal levels of phosphate. MOPS growth media consisted of MOPS Salts, 0.2% Casein Amino Acids, 0.2% Glucose, and 1.32 mM KH_2PO_4. When necessary, chloramphenicol and/or kanamycin were added to the growth media at a concentration of 33 ug/mL and 25 ug/mL respectively. To induce gene expression from the ASKA

collection, IPTG was added to growth media at a concentration of 0.1 mM. Cell strains used for protein expression were grown in 2xYT rich media containing 33 μg/mL of chloramphenicol.

The details for all E. *coli* strains are described in Table A.1. The *arsC* gene from both Jude 1 and DHB4 was replaced with a kanamycin resistance marker using the Wanner gene knockout method as previously described[177]. The following genetic elements were also knocked out of the chromosome of the DHB4 Δ*arsC* mutant strain: *gstB* and the *arsRBC* operon. Finally, the *arsC* gene was also removed from the mutant strain DHB4Δ*grxA*Δ*grxB*Δ*grxC*. For each subsequent knockout, the kanamycin resistance marker was first removed using the pcp20 plasmid[177].

The details for all E. *coli* constructs are described in Table A.2. Genetic selection was performed using the library of ASKA clones where the GFP tag had been previously removed, leaving behind a peptide scar sequence of GLCGR. We removed the C-terminal Scar from pGstB using Quick change PCR. In addition, using quick change, we made variants of GstB with mutations in selected reasons. Specifically, we made three variants in which the CXXC motif was selectively mutated to alanine: pGstB$_{C134A/C137A,}$ pGstB$_{C137A}$, and pGstB$_{C134A.}$ An additional three mutants were constructed in which the purported essential arginine residues were mutated to glutamine: pGstB$_{R111Q/R119Q,}$ pGstB$_{R111Q}$, and pGstB$_{R119Q.}$ Finally, for GstB crystallization, we constructed another variant pGstB$_{P\text{-}WT}$ where a Factor XA protease site IDGR was added downstream of the His-Tag and upstream of the start codon using overlap extension PCR in the event it was necessary to remove the His-tag.

Genetic selection

Electrocompetent Jude 1 *E. coli ΔarsC* mutant strain, EQ217, was transformed with the ASKA library. Transformed cells were rescued in LB media containing 0.1 mM IPTG for two hours. After rescue, cells were plated on MOPS plates containing Chloramphenicol, Kanamycin, 0.4 mM Sodium Arsenate, and 0.1 mM IPTG. Plated cells were allowed to grow for two days at 37^{o} degrees Celsius. After two days, colonies were selected and their corresponding pCA24N constructs from the ASKA library were sequenced.

Growth curves

Selected cell strains were grown overnight in MOPS 1X Minimal Media; Chloramphenicol was added to the media of strains containing a pCA24N construct. The following day, strains were sub-cultured at a dilution of 1:100 into 24-well growth culture plates. The total volume of growth media in each well was 20% the total well-volume. 0.1 mM IPTG and varied concentrations of arsenate from 0 to 2 mM was added to each well. Cell growth was monitored for 24 hours at 37°C using a Synergy HT plate reader at an absorbance of 600 nm. Growth curves were fit to a Gompertz hyperbolic model[178]. From the model we extracted doubling time, lag time, and saturation absorbance.

Protein expression and purification

Both GstB and the GstB inactive variant were purified, as previously described[175], from the pGstB and pGstB$_{R111Q/R119Q}$ constructs, respectively. Cells were grown overnight in 2XYT media containing chloramphenicol. The next day, cells were sub-cultured at a dilution of 1:100 in a 500 mL flasks containing 2XYT. Once cells reached an OD of 0.8-1, they were induced with 0.5 mM IPTG for 3 hours at 37°C. After induction, cells were spun down for 20 minutes and the pellet was stored overnight at -20°C.

Cell pellets were resuspended in 30 mL of buffer A containing 50 mM HEPES pH 7.5, 40 mM imidazole, 2 mM DTT, and 300 mM NaCl. Cells were lysed by passage twice through a French Press; debris from lysed cells was sedimented by centrifugation for 30 minutes. GstB was purified from cell lysate using Nickel NTA resin. Non-specific binding to the column was washed using buffer B, which contained Buffer A and 65 mM imidazole. GstB was eluted in a solution containing Buffer A and 250 mM imidazole. Purified GstB protein was dialyzed overnight in 4 L of dialysis buffer containing 50 mM HEPES pH 7.5, 10% glycerol and 10 mM NaCl.

Measurement of Arsenite in solution

Solutions of arsenite were separated from arsenate using disposable Strata SAX anion exchange columns. Arsenate, which is negatively charged at a pH > 2.2 binds to the positively charged column, whereas arsenite which is uncharged at pH < 9.2 will not bind to the column. Therefore, when a solution containing a mixture of arsenate and arsenite is passed through the column, purified arsenite can be collected in the flow through. The levels of arsenite can then be semi-quantitated using the EZ arsenic detection kit provided by HACH chemicals. This kit detects levels of arsenic as low as 50 ppb. Total arsenic is detected by the zinc catalyzed reduction of arsenite to arsine gas which is then captured by mercuric bromide strips causing a color change. Strips that appear dark orange/black indicate levels of arsenic above 500 ppb.

Measurement of Arsenite accumulation in-vivo

Cell strains grown overnight were sub-cultured at a dilution of 1:100 at 37°C in MOPS media containing the required antibiotic for each respective strain. When cells reach an OD of 0.3, IPTG was added to the growth media at a concentration of 0.1 mM.

Once cells reached an OD of 0.6-0.8, 1 mM Arsenate was added to each strain and growth was continued for one hour. After one hour of growth, all cell strains were normalized to the same OD, and 5 mLs of cells were pelleted for 10 minutes. The solution was diluted using 45 mLs of water and the amount of arsenite present in each sample was measured using the assay described above.

In-vitro arsenate reduction

The reaction buffer for in-vitro arsenate reduction was 100 mM HEPES at neutral pH 7. In the non-catalyzed reaction, 1 mM of GSH was incubated with 50 mM sodium arsenate. For the enzyme catalyzed reaction, purified enzyme was added to the mixture at a concentration of 0.12 mM. The total reaction volume was 500 μL and samples were incubated for 60 minutes total. After 30 minutes of incubation, arsenite production was measured using half of the mixture; after 60 minutes, arsenite production was measured using the remaining half of the mixture. At each time point, the mixture was diluted to 3mLs, the pH was subsequently lowered to 4, and, finally, the mixture was separated using Strata-SAX anion exchange columns. The reaction mixture was lowered to 4 to separate any As(III)-GSH conjugates known to freely formed in solution. After separation, the flow-through was diluted to a volume of 10 mLs using water before arsenic detection. Arsenite in each reaction mixture was semi-quantitated using the arsenite detection assay described above.

GstB bromoacetate activity

The assay for bromoacetate activity was modified slightly from that described previously[175]. Bromoacetate activity was quantified by monitoring the consumption of reduced GSH. HEPES buffer at 100 mM and pH 7 served as the reaction buffer and 35

mM bromoacetate and 5 mM GSH was added to the mixture. To initiate the reaction, purified GstB was added to the reaction mixture at a concentration of 0.5 μM. GSH consumption was compared to no enzyme controls. The substrates were reacted for varied amounts of time, at which point, the reaction was quenched using 30 mM monobromobimane which conjugates bimane to any remaining reduced GSH. The remaining GSH conjugated to bimane was quantitated by measuring the absorbance at 338 nm using HPLC analysis. Reaction rate kinetics was obtained after subtracting the amount of non-catalyzed Glutathione-S-acetate formed.

GstB activity via NADPH coupled assay

GstB activity for arsenate reduction was quantified using a similar protocol used for arsC activity, in which activity is measured using a NADPH coupled assay[169]. Briefly, each 100 μL reaction mixture contained the following: 0.3 mM of NAPDH, 1 μg/mL glutathione reductase, and 1 mM reduced GSH. If indicated, concentrations of As(V) varied between 20 to 150 mM, and purified enzyme was added to the master mixture above at a concentration of 0.12 mM. If indicated, Grx2 was added to the master mixture at a concentration of 10 μg/mL. Reaction kinetics were obtained by measuring the decrease in absorbance at 340 nm using a Synergy HT plate reader. [NAPDH] was determined using its extinction coefficient at 340 nm of $6200 M^{-1} cm^{-1}$ and 0.3 cm path length corresponding to the depth of 100 μL reaction in 96 well plates.

Results

GstB confers arsenate resistance to E.coli ΔarsC

EQ217, a *ΔarsC E. coli* strain derived from DH10B, is hypersensitive to arsenate with complete colony growth inhibition observed on MOPS minimal media agar plates containing 0.4 mM sodium arsenate. EQ217 was transformed with the ASKA library[172] and transformants were plated on MOPS minimal media agar plates with 0.4 mM sodium arsenate (NaH_2AsO_4). Several colonies of varied size arose after two days of incubation at 37° Celsius. Sequencing showed that large colonies encoded for wild type arsC from the ASKA vector. Interestingly, the majority of small colonies randomly picked for sequencing were found to encode the glutathione-s-transferase enzyme GstB. A small number of colonies that were sequenced also encoded for the colonic acid biosynthesis protein cpsB, and the transcription factors dksA and fruR. To reconfirm that arsenate resistance was caused by gene overexpression, plasmids identified from the genetic screen were retransformed in EQ276, a DHB4 *ΔarsC* mutant strain. Only multi-copy expression of the *gstB* gene could confer growth on MOPS plates containing more than 1 mM sodium arsenate (Figure A.2 A). Further investigation showed that DHB4 *ΔarsC* mutants transformed with GstB formed good size colonies on plates containing up to 3 mM NaH_2AsO_4, and even formed tiny colonies on plates containing 5 mM sodium arsenate (Figure A.2 B). Every gene expressed from the ASKA collection contains a C terminal linker sequence, GLCGA. Removal of this scar from GstB did not affect arsenate resistance in EQ276 mutant cells expressing GstB. The results of this genetic screen suggested that arsenate may serve as an electrophilic substrate for this glutathione-s-transferase.

Expectedly, both ArsC and GstB mediated resistance are dependent on glutathione. Multicopy expression of GstB or ArsC in glutathione deficient cell strains (WP758) could not rescue growth in media containing 0.1 mM sodium arsenate (data not shown). In addition, the arsenate reductase enzyme is dependent on glutaredoxins for the reduction of the ArsC-arsenate-glutathione intermediate complex into ArsC-arsenite and a glutathione-glutaredoxin mixed disulfide[168]. At concentrations above 0.1 mM NaH_2AsO_4, *E.coli* mutant CC109, carrying a deletion of glutaredoxins A, B, and C as well as *ΔarsC,* showed increased sensitivity to arsenate relative to the parental strain EQ279 (DHB4Δ*arsC)*. Transformation of mutant strain CC109 with pArsC could not restore healthy growth on plates containing low millimollar concentrations of arsenate (Figure A.2 B). In contrast, CC109 cells transformed with pGstB showed satisfactory growth up to 3 mM arsenate.

Thus, unlike ArsC, resistance to arsenate by GstB is not dependent on the action of the *E.coli* glutaredoxins.

ArsC and ArsB together with the trans-acting repressor ArsR comprise the arsenate resistance operon *arsRBC*. ArsC reduces arsenate to arsenite which is then exported from the cell by the ArsB pump [179]. DHB4 containing a complete deletion of the Δ*arsRBC* operon, CC112, could not grow on plates containing more than 0.3 mM arsenate. This phenotype could not be rescued by multi-copy expression of ArsC or GstB (Figure A.2 C).

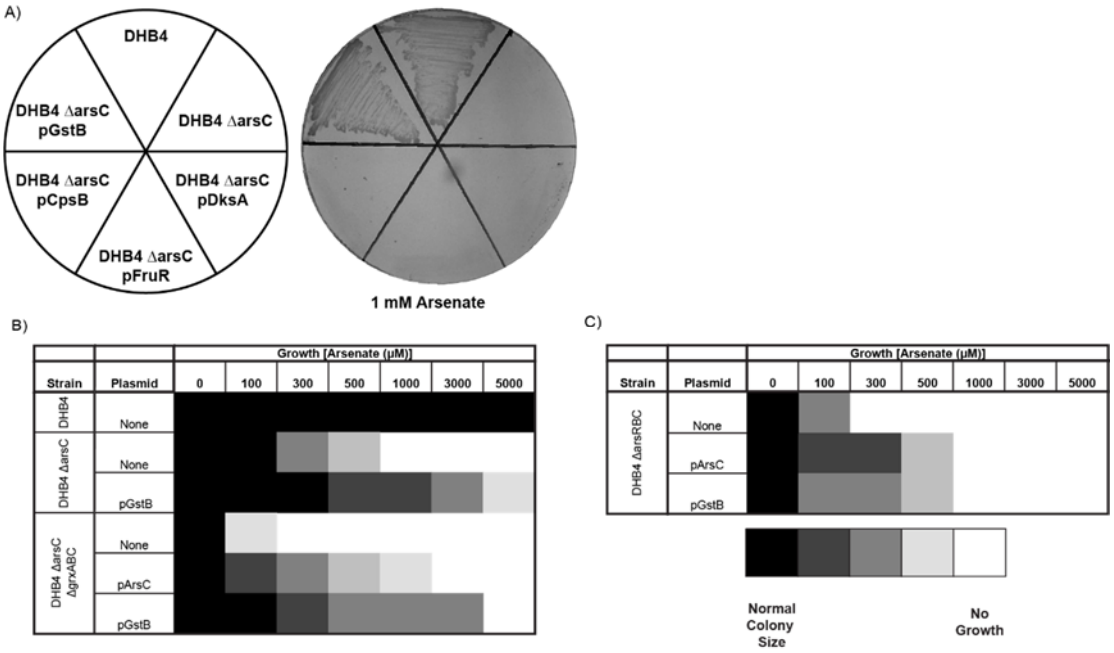


Figure A.2: GstB overexpression confers As(V) resistance in Δ*arsC* knockout mutants.

(A) Only selected Δ*arsC* mutants overexpressing gstB from the genetic screen survived on agar plates containing more than 1 mM arsenate. (B) Selected mutant strains were grown on agar plates containing varied arsenate concentrations. GstB overexpression confers growth on plates containing between 3-5 mM arsenate. The growth of glutaredoxin null mutants (Δ*grxABC*) overexpressing GstB indicates that GstB conferred resistance is not dependent on glutaredoxins. (C) Both ArsC and GstB conferred resistance are dependent on the ArsB arsenite efflux pump

Considering that arsR is only required for regulation of ArsC and ArsB transcription, this evidence suggests that, in vivo, the arsenite transporter is essential for arsenate detoxification facilitated by GstB.

Residues essential for GstB activity

GstB contains a canonical thioredoxin-like CXXC motif comprising Cys134 and Cys137. A mutant GstB enzyme in which both cysteine residues had been mutated to alanine displayed full arsenate detoxification ability *in-vivo*. Specifically, DHB4 Δ*arsC* expressing the double cysteine knockout mutant, $GstB_{C134A/C137A}$, grew in the presence of 1 mM sodium arsenate with a doubling time of 200 +/- 10 min compared to 205 +/- 10 min, for cells expressing wild type GstB. (Figure A.3 and Supplementary Figure A.7).

Our efforts to crystallize *E.coli* GstB led to poor quality crystals that diffracted at <3 angstroms but were not solvable due to crystal heterogeneity. Thus, we referred to the recently reported crystal structure of a GstB homolog from *Salmonella enterica*, YliJ, in complex with GSH [180] (PDB 4KH7). YliJ shares 83% amino acid identity with the E.coli GstB, but contains only one of the two cysteines, Cys137, which constitute the CXXC motif. Consistent with our finding that neither cysteine residue affected arsenate detoxification, the crystal structure shows that the homologous cysteine residue in YliJ is distant from both the GSH and the putative electrophilic binding site (Figure A.3 A). We observed that in the crystal structure of YliJ three conserved arginine residues, R7, R111, and R119, form a patch of positive charge in close proximity near the glutathione binding site. The arginine closest to glutathione, R7, may be involved in glutathione binding. We hypothesized that the two distal arginine residues, R111 and R119, found within the electrophilic binding domain of GstB might have a role in positioning anions and electrophiles in close proximity to the GSH electron donor. Therefore, three GstB variants,

$GstB_{R111Q}$, $GstB_{R119Q}$, and $GstB_{R111Q/R119Q}$, containing selective mutations of the two arginine residues were constructed. Unlike $GstB_{C134A/C137A}$, removal of each individual arginine residue decreased GstB conferred resistance in the arsenate sensitive strain EQ276. Removal of both arginine residues from GstB, $GstB_{R111Q/R119Q}$, abolished arsenate resistance in EQ276 (Figure A.3 and Supplementary Figure A.7).

We analyzed whether loss of activity caused by mutagenesis was due to protein instability and aggregation. Both GstB and $GstB_{R111Q/R119Q}$ were expressed and purified to near homogeneity using His-tag purification. Size exclusion chromatography showed an identical elution pattern for both wild type GstB and $GstB_{R111Q/R119Q}$ (Supplementary Figure A.8). Differential scanning fluorimetry revealed that the $GstB_{R111Q/R119Q}$ mutant

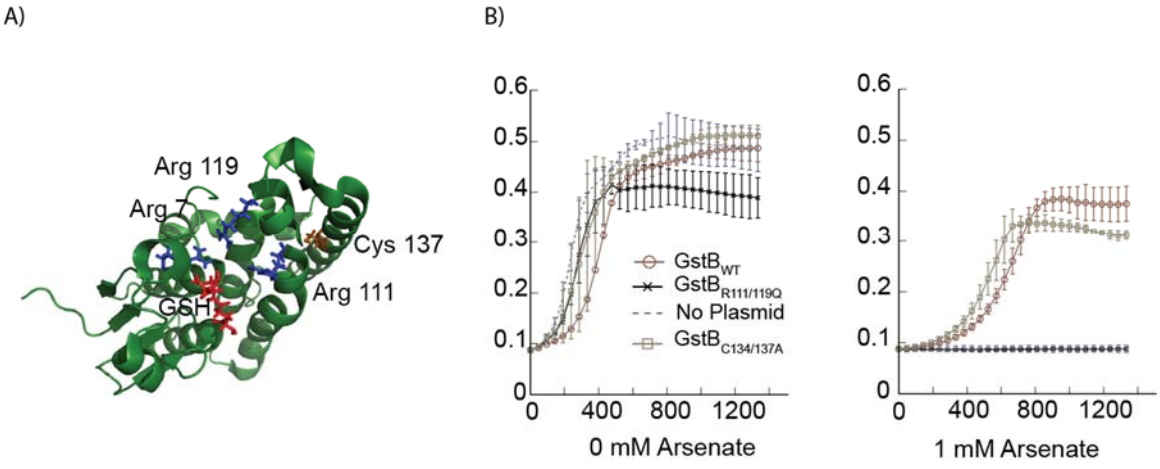


Figure A.3: GstB residues Arginine-111 and Arginine-119 are essential for arsenate resistance

(A) Crystal structure of the *E. coli* GstB homolog in Salmonella (PDB 4KH7). Three arginine residues (blue stick models), R-7, R-111, and R-119, form the putative electrophilic binding site near GSH (red stick model). (B) Growth of *ΔarsC* mutants expressing either wild type GstB variant ($GstB_{WT}$), GstB variant ($GstB_{R111Q/R119Q}$), or wild type GstB variant ($GstB_{C134A/C137A}$) grow in liquid media containing 0 mM arsenate (left) or 1 mM arsenate (right)

even exhibit a slightly increased stability to thermal unfolding (47^{o} +/- 0.2^{o}C) relative to the wild type enzyme (44^{o} +/- 0.4^{o} C) (Supplementary Figure A.10). Finally, we measured whether this GstB variant retained its dehalogenation activity for bromoacetate[175]. In vitro activity for bromoacetate was monitored by the measuring the decrease in reduced GSH upon the addition of bromoacetate and GstB. Notably, wild type GstB displayed activity towards bromoacetate as previously reported, while $GstB_{R111Q/R119Q}$, showed no activity towards bromoacetate when compared to the non-enzymatic reaction (Supplementary Figure A.11).

GstB catalyzes the reduction of As(V) to As(III)

The finding that GstB overexpression does not confer arsenate resistance in Δ*arsRBC* cells lacking the arsenite transporter indicates that this enzyme plays a role in the reduction of AS(V) to As(III). We used a straightforward two-step assay to detect the accumulation of As(III) in complex biological mixtures such as culture supernatants or cell lysates. This method could distinguish arsenite concentrations varying from 0.05 to 0.5 mM (Supplementary Figure A.9). First, As(III) is separated from As(V) in solution using anion exchange. Arsenate, negatively charged at a pH above 2.2, will stay bound to the column whereas unionized arsenite passes through the flow through. The anion exchange columns were able to separate arsenite from solutions containing 50 mM sodium arsenate (Supplementary Figure A.9). Separated As(III) in flowthrough is measured using a colorimetric assay sensitive to less than 25 ppb total arsenic. The colorimetric assay reduces all arsenic species into arsine gas which subsequently comes into contact with strips containing mercuric bromide. Reaction between arsine gas and mercuric bromide discolors the test strip proportionally to the concentration of total arsenic in solution[181].

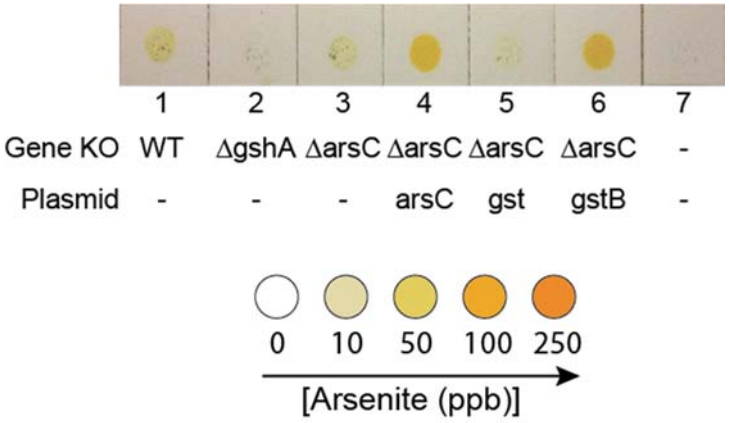


Figure A.4: GstB overexpression results in the reduction of arsenate to arsenite in-vivo

Colorimetric tests indicate the level of arsenite accumulation in the supernatant of mutant strains cultured in liquid media containing 1 mM As(V). Detection is sensitive to arsenite, not arsenate, in the supernatant (Panel 7). Glutathione null mutant strains cannot reduce As(V) to As(III) (Panel 2). ArsC null mutants overexpressing GstB export significant amounts of As(III) (Panel 6); As(III) production is specific to GstB expression and not general GST expression (Panel 5).

We measured the levels of arsenite that accumulated in supernatant of select E. *coli* strains after incubation with 1 mM sodium arsenate. Protein expression was induced in mid-exponential cells growing in MOPS minimal media using IPTG. Arsenate was added to the media one hour after induction. After 1 hour of incubation with arsenate the cells were pelleted and arsenite levels in the supernatant was determined as described above (Figure A.4). No arsenite was detected in the *E. coli* mutant, WP758, incapable of synthesizing GSH. A very low level of arsenite was detected in the supernatant of the *E.coli* Δ*arsC*Δ*gstB* strain EQ301. This low level of arsenite is presumably due to the slow, non-enzymatic reduction of As(V) by intracellular GSH[182]. The presence of arsenite in

this strain as compared to WP758 supports data in which both EQ279 and EQ301 grew on plates containing up to 0.4 mM sodium arsenate while WP758 would not grow on plates containing 0.1 mM sodium arsenate.

A low concentration of arsenite in the culture supernatant was also detected in DHB4 cells. This was presumably because exposure to arsenate for 1 hour is not enough time to allow a sufficient induction of ArsC synthesis. In contrast, ArsC expression induced from the multicopy plasmid exhibited significant accumulation of extracellular arsenite. A similar result was observed in cells expressing GstB indicating that the expression of this glutathione S-transferase directly or indirectly mediates the catalytic reduction of arsenate into arsenite (Figure A.4, Panel 6). Finally overexpression of the better studied *E. coli* glutathione S-transferase Gst in EQ301 did not result in significant arsenite accumulation (Figure A.4, Panel 5).

We next sought to evaluate whether GstB directly catalyzes arsenate reduction. The ability of GstB to reduce arsenate was tested using two separate methods. First, accumulation of arsenite was directly measured from in-vitro assays containing purified GstB, reduced GSH, and sodium arsenate. Incubation of 0.12 mM purified GstB in these assays significantly increased the levels of arsenite after both 30 and 60 minutes. Conversely, incubation with 0.12 mM purified $GstB_{R111Q/R119Q}$ did not increase the levels of arsenite produced as compared to the non-enzymatic reaction (Figure A.5 A). Second, the rate of arsenate reduction in the presence of GstB was monitored using a redox coupled assay containing NADPH, GOR, GSH, and arsenate[169]. The redox couple assay also showed that the rate of arsenate reduction was significantly higher in the presence of GstB, but not in the presence of equal amounts of $GstB_{R111Q/R119Q}$ (Figure A.5 B). The addition of purified Glutaredoxin, Grx2, did not improve GstB activity in-vitro (data not shown).

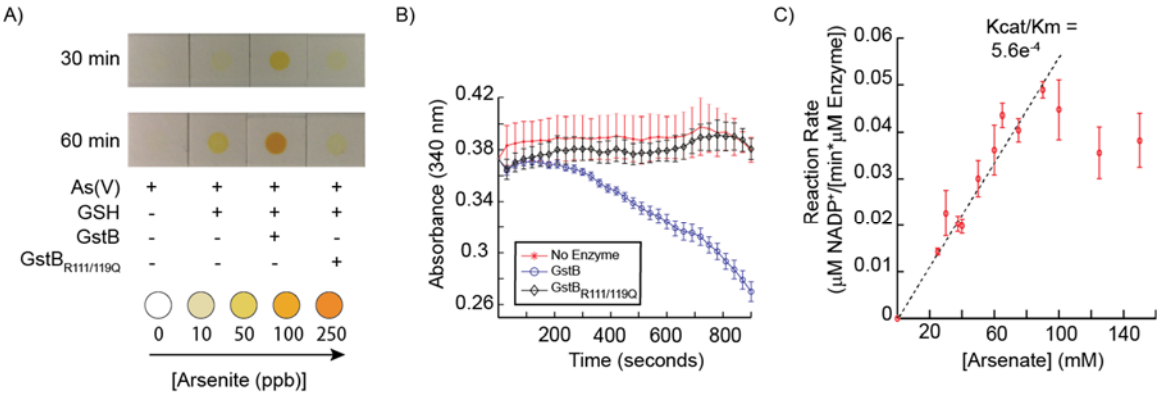


Figure A.5: GstB reduces arsenate to arsenite in-vitro

(A) Time-course assay for in-vitro conversion of arsenate to arsenite under various reaction conditions. Mixture of 0.12 mM of active GstB with 50 mM As(V) and 1 mM GSH results in significant As(III) accumulation (Panel 3) as opposed to no enzyme (Panel 2) or inactive $GstB_{R111/119Q}$ (Panel 4) (B) NADPH oxidation assay for arsenate reduction in the presence of 0.12 mM active GstB, 50 mM As(V), and 1 mM GSH. (C) Reaction kinetics of GstB activity for arsenate using NADPH oxidiation assays. K_{cat}/K_m is approximated to be $5.6e^{-4}$.

Finally, GstB activity for arsenate was assayed under saturating glutathione concentrations and concentrations varying from 20 mM to 150 mM sodium arsenate. Activity was found to be linear at concentrations below 100 mM Arsenate. Therefore, fitting the kinetic data at concentrations below 100 mM, we estimate that GstB has a kcat/km of 0.6 $(min*M)^{-1}$ with respect to arsenate reductase activity (Figure A.5 C).

DISCUSSION

By using a genetic screen to identify suppressors of arsenate sensitivity caused by the deletion of the native *arsC* arsenate reductase gene, we identified a glutathione-s-transferase, GstB, that when overexpressed conferred high levels of arsenate resistance.

Both *in-vivo* and *in-vitro* assays investigating GstB activity, show that this enzyme is capable of facilitating the reduction of arsenate into arsenite. *In-vivo*, we find that arsenate resistance is still dependent upon the presence of ArsB and that overexpression of GstB is linked to the accumulation of arsenite in the supernatant. *In vitro*, we provide direct evidence indicating that GstB accelerates arsenate reduction. Given this evidence, we believe that GstB could support arsenate reduction via two proposed reaction mechanisms that do not involve glutaredoxins (Figure A.6).

In one such schema, GstB could facilitate a rate limiting step in which reduced glutathione is first conjugated to arsenate. Subsequently, another glutathione molecule in solution could spontaneously react with the arsenate-glutathione intermediate complex to form oxidized glutathione and thereby reduce arsenate to arsenite. Another plausible mechanism given our evidence is that first reduced glutathione conjugates to arsenate in solution; GstB then facilitates the reduction of this intermediate complex via glutathione bound to the transferase. While we cannot exclude either mechanism, we believe that the former mechanism is more plausible for two reasons. First, steric collisions around the putative GSH binding site of GstB would greatly hinder the ability of a pre-formed GSH-arsenate complex to interact with the second equivalent of glutathione bound to GstB. Second, a nuclear magnetic resonance study of the reaction between glutathione and arsenate, *in vitro*, could not find evidence of an arsenate-GSH complex[182]. Based on previous ArsC studies, it is known that such arsenate-thiol complexes form as intermediate species in ArsC[168]. The absence of an arsenate-GSH complex in the NMR study suggests that the rate limiting step of non-enzymatic arsenate reduction with GSH is the putative formation of the arsenate-GSH complex, and that the subsequent binding of a second equivalent of GSH and reduction is rapid. As such, if the later mechanism were

correct, GstB would not accelerate the rate of the non-enzymatic reaction as the reduction step is not rate limiting. Therefore, it is most likely that GstB catalyzes the formation of an arsenate-GSH conjugate, which then leaves the enzyme and encounters a second equivalent of GSH, resulting in the reduction of arsenate to arsenite (Figure A.6 Route A). Given the relatively high concentration of GSH in the cytoplasm, millimolar levels, a spontaneous encounter between a free GSH and the arsenate-GSH complex would likely be rapid.

To our knowledge GstB is the first known bacterial GST with arsenate reductase ability. However, GstB is not the first instance in which a glutathione-s-transferase has been shown to catalyze arsenate reduction[183]. In fact, the human monomethylarsenate reductase enzyme, hGSTO1 (PDB 1eem) or MMAV reductase, belongs to the Omega class of GSTs [184], [185]. Furthermore, in *L. major*, the glutathione-s-transferase like enzyme,

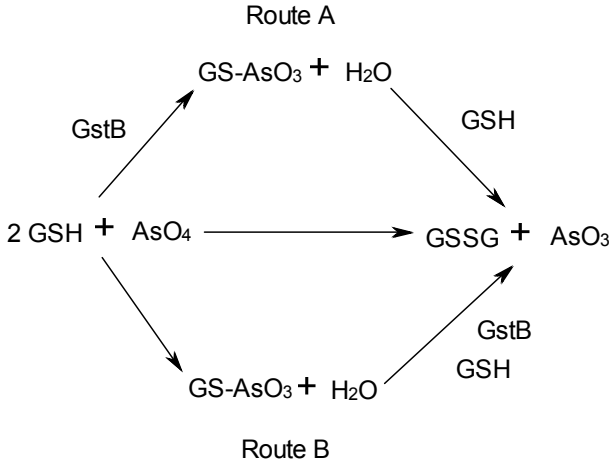


Figure A.6: Proposed Mechanisms of GstB conferred As(V) resistance

Route A. GstB increases the rate of non-enzymatic reduction of As(V) via reduced GSH. *Route B.* GstB increases the rate of GSH-As(V) conjugation. Once formed, the intermediate undergoes spontaneous reduction to As(III) and oxidized GSH (GSSG).

TDR1 (PDB 4AGS), which shows 29% similarity to hGSTO1 was also shown to have reductase activity for both MMAV and As(V)[186], [187, p. 1]. Protein alignment of E. *coli* GstB to both enzymes shows 26% and 28% protein sequence identity to hGSTO1 and TDR1, respectively. One interesting difference between GstB and both TDR1 and hGSTO1 is that the later eukaryotic reductase enzymes contain a catalytic cysteine residue [184], [186]. Conversely, we did not find any evidence that the cysteine residues in GstB contributed significantly to arsenate resistance. This difference in the catalytic site may suggest why GstB shows very poor kinetics for arsenate reduction, and it may be of future interest to insert a cysteine into the active site of GstB to elucidate its effect on kinetics. Therefore, GstB serves as an intriguing starting point for future directed evolution experiments. It would be interesting to reveal what essential mutations in the active site, such as a cysteine, would be necessary to drastically improve its activity towards arsenate, and whether these mutations also result in a loss of activity for bromoacetate. Our data once again shows the plasticity of arsenate detoxification in biology and that novel pathways for arsenic chemistry may still be yet discovered

Strain Name	Parent Strain	Donor Strain	Chromosomal Mutations	Resistance	Source or Reference
Jude1 DH10B					Lab Collection
DHB4					Lab Collection
EQ279	DHB4		ΔarsC	Kanamycin	This study
EQ217	JUDE1		ΔarsC	Kanamycin	This study
EQ301	EQ279		ΔarsCΔgstB	Kanamycin	This study
grxA750(del)::kan	JW0833		grxA750(del)::kan	Kanamycin	Keio collection
grxB734(del)::kan	JW0833		grxB734(del)::kan	Kanamycin	Keio collection
grxC722(del)::kan	JW0833		grxC722(del)::kan	Kanamycin	Keio collection
CC101	DHB4	grxA750(del)::kan	ΔgrxA	Kanamycin	This study
CC102	DHB4	grxB734(del)::kan	ΔgrxB	Kanamycin	This study
CC103	DHB4	grxC722(del)::kan	ΔgrxC	Kanamycin	This study
CC104	CC101	CC103	ΔgrxAΔgrxC	Kanamycin	This study
CC105	CC104	CC102	ΔgrxAΔgrxBΔgrxC	None	This study
CC106	EQ279		ΔarsCΔgrxA	Kanamycin	This study

Table A.1: Strains used in this study

CC107	EQ279	ΔarsCΔgrxB	Kanamycin	This study
CC108	EQ279	ΔarsCΔgrxC	Kanamycin	This study
CC109	CC105	ΔarsCΔgrxAΔgrxBΔgrxC	None	This study
CC110	EQ279	ΔarsCΔarsB	Kanamycin	This study
CC111	DHB4	ΔarsB	Kanamycin	This study
CC112	DHB4	ΔarsRΔarsBΔarsC	Kanamycin	This study
WP758	DHB4	DHB4gshA20	Kanamycin	Lab Collection

Table A.1 continued

Plasmid	Construct Derived From	Source
pGstB$_{ASKA}$	-	ASKA collection
pGst$_{ASKA}$	-	ASKA collection
pArsC$_{ASKA}$	-	ASKA collection
pCpsB$_{ASKA}$	-	ASKA collection
pFruR$_{ASKA}$	-	ASKA collection
pDksA$_{ASKA}$	-	ASKA collection
pGstB$_{WT}$	**pGstB$_{ASKA}$**	This study
pGstB$_{C134137A}$	**pGstB$_{WT}$**	This study
pGstB$_{R111/1119Q}$	**pGstB$_{WT}$**	This study
pGstB$_{homologue}$		This study
pGstB$_{P\text{-}WT}$	**pGstB$_{WT}$**	This study
pGstB$_{P\text{-}R111/119Q}$	**pGstB$_{R111/1119Q}$**	This study

Table A.2: Plasmids used in this study

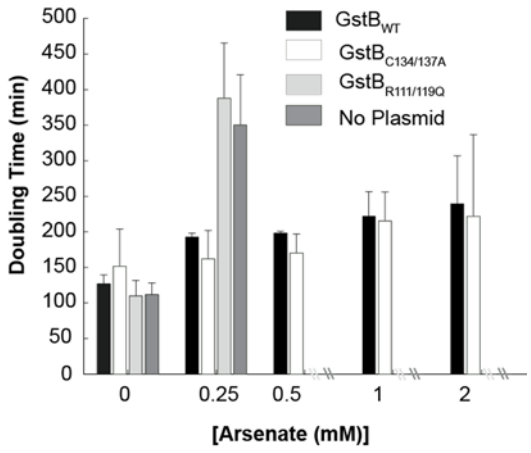


Figure A.7: Arsenate resistance conferred by GstB mutant variants

Growth rate analysis of DHB4 Δ*arsC* mutant strain expressing selected GstB variants. Analyzed growth in liquid media containing varied concentrations of sodium arsenate.

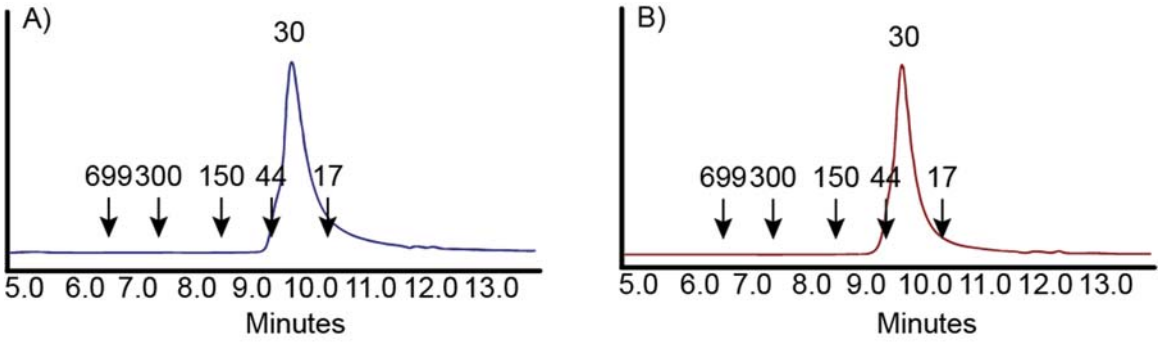


Figure A.8: Size exclusion of enzyme variants

[A] GstB$_{P\text{-}NS}$ [B] GstB$_{P\text{-}R111/119Q}$. Arrows represent retention time of protein standards.

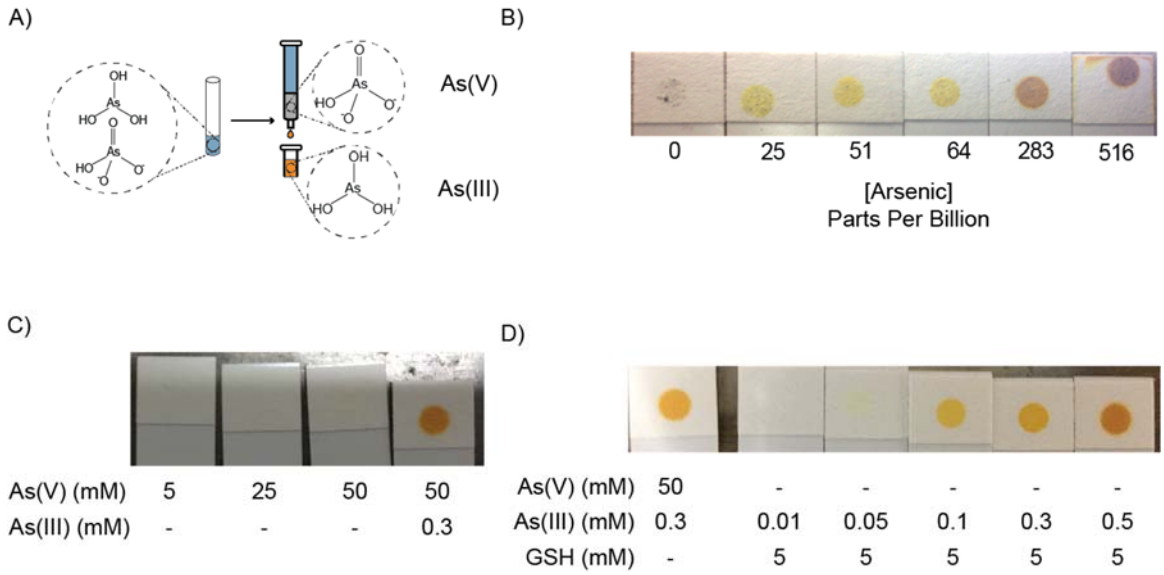


Figure A.9: Two-step semi-quantitative assay for As(III) in solution

A)Separation of As(III) from As(V) in solution using anion exchange. As(III) comes through the flow through. B) Sensitivity of colorimetric assays for arsenic. Strips are semi-quantitative for arsenic concentrations between 20 to 500 ppb. C) Assay can separate low levels of arsenite (As(III)) from solutions containing as much as 50 mM arsenate (As(V)). D) Separation of As(III) from As(V) at pH 4. Colorimetric assays can distinguish between varied concentrations of arsenite in solution before separation.

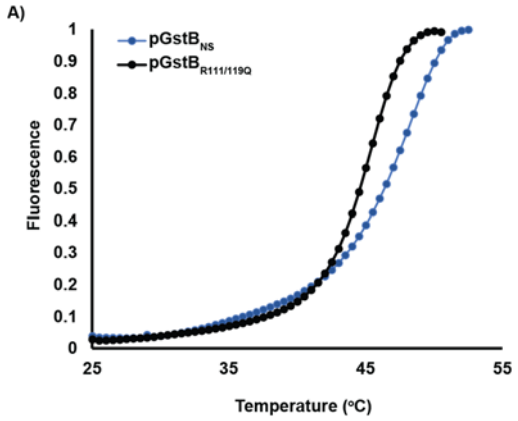


Figure A.10: Differential Scanning Fluorometry analysis of GstB active and inactive variant

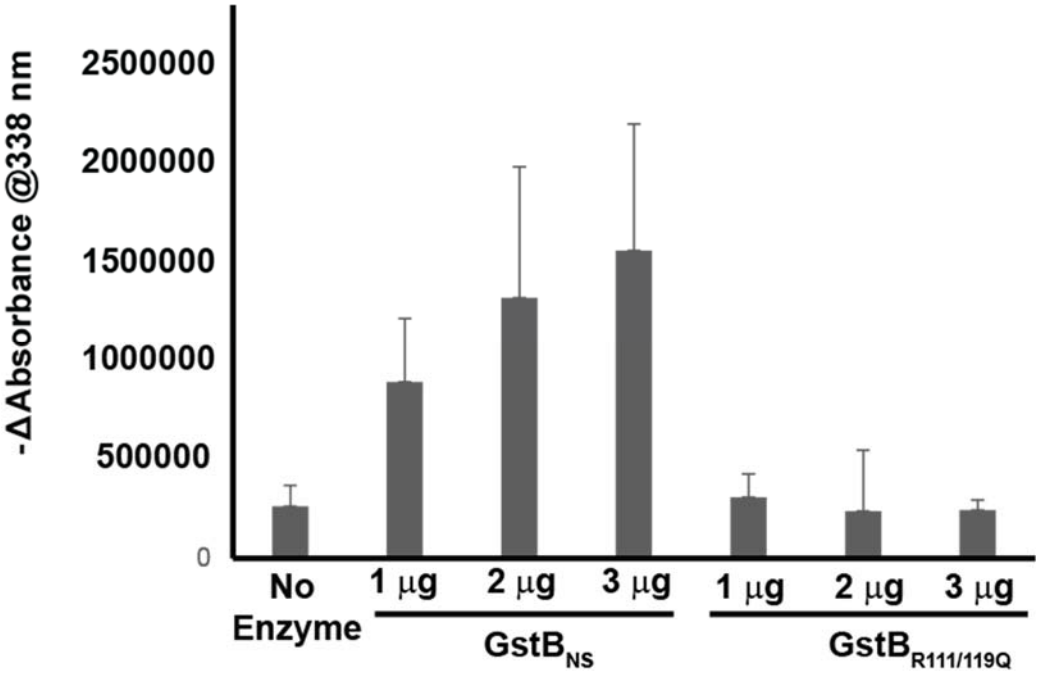


Figure A.11: Analysis of in-vitro activity of GstB$_{R111Q/R119Q}$

Analysis of bromoacetate acivity with respect to active variant GstB and inactive variant GstB$_{R111/119Q}$. Reaction was monitored by the reduction in reduced GSH over time. Reduced GSH remaining at the end of the reaction was conjugated to monobromobimane which absorbs at 338 nm

Appendix B

TRUE DIVERSITY AND DIVERSITY INDEX

In ecology, measurements of diversity are referred to as diversity indices. The simplest type of diversity index is species richness which reports the total number of species, S, found within a population. A more accurate representation of diversity is the diversity index, HCDT entropy, which is a function of both species frequency, f_i, and the diversity order, q, [Equation B.1][106].

$$x(q) = \frac{(1 - \sum_{i=1}^{S} f_i^q)}{(q-1)} \quad \text{[Eq B.1]}$$

Where:

x = HCDT entropy

q = The diversity order

S = The total number of unique species

f_i = The frequency of the current species *i*

In Equation B.1, the diversity order, q, defines the diversity index's sensitivity to species frequency. For example, analogously to species richness, when using a diversity order where q=0, the diversity index ($x=\Sigma(f_i)^0-1= S-1$) only accounts for every unique species, S, found within a population, but disregards specie's frequency (f_i). As a general rule, diversity indices where order $q<1$ tend to give more weight to the diversity contributed by rare species. Diversity indices where order $q>1$ tend to favor species found at a high frequency within the population and ignore diversity contributed by rare species. Because of this bias, the arguably fairest measurement of a diversity index is where q = 1 and the frequency of both rare and common species is weighted proportionately[106]. Equation B.1 is undefined for q = 1, but its limit at q =1 exists and can be expressed as shown in

Equation B.2. In fact, when q = 1, the diversity index is identical to Shannon entropy estimation of diversity[106], [188], [189].

$$\lim_{q\to 1} x(q) = -\sum_{i=1}^{S} f_i \ln(f_i) \text{ [Eq B.2]}$$

Where:

x = The diversity index (Equal to shannon entropy)

q = The diversity order

S = The total number of unique species

f_i = The frequency of the current species i

Each diversity index described above (species richness, HCDT entropy, and Shannon entropy) is a monotonic function that is proportional to the diversity of a population. In other words, the larger the index then the more the diverse the population. More importantly, we assume that populations with the same measurement of diversity index have equal diversity. However, one caveat is that diversity indices do not scale linearly with the population diversity. A population with a Shannon diversity index of 10 is not twice as diverse as a population with a Shannon diversity index of 5. In order to compare diversity across populations, the diversity index has to first be normalized such that the values for diversity will scale linearly. The normalized form of a diversity index is referred to as "true diversity"[106]. The true diversity is found by noting that diversity does scale linearly if the species within the population are represented at an equal frequency. For example, a population containing twenty species represented at an equal frequency (5%) is twice as diverse as a population containing ten species represented at an equal frequency (10%). Consequently, the true diversity of a population is calculated by equating its diversity index to a theoretical population where every species is represented

at the same frequency [Equation B.3]. The calculated number of species, or "effective number of species", in this theoretical population represents the "true diversity". In contrast to diversity index, a population with a true diversity of 10 is twice as diverse as a population with a true diversity of 5.

Equate diversity index to theoretical population of equally distributed species:

$$\frac{(1-\sum_{i=1}^{S} f_i^q)}{(q-1)} = x(q) = \frac{(1-\sum_{i=1}^{D_T} (1/D_T)^q)}{(q-1)}$$

$$\sum_{i=1}^{S} f_i^q = D_T * (D_T)^{-q} = D_T^{1-q}$$

$$(\sum_{i=1}^{S} f_i^q)^{1/(1-q)} = D_T(q) \text{ [Eq B.3]}$$

Where:

D_T = "True diversity"/"effective number of species"

x = The diversity index

q = The diversity order

S = The total number of unique species

f_i = The frequency of the current species *i*

Primer Name	Primer Sequence
YarivH-FOR 1	GTT ATT GCT AGC GGC TCA GCC GGC AAT GGC GGA KGT RMA GCT TCA GGA GTC
YarivH-FOR 2	GTT ATT GCT AGC GGC TCA GCC GGC AAT GGC GGA GGT BCA GCT BCA GCA GTC
YarivH-FOR 3	GTT ATT GCT AGC GGC TCA GCC GGC AAT GGC GCA GGT GCA GCT GAA GSA STC
YarivH-FOR 4	GTT ATT GCT AGC GGC TCA GCC GGC AAT GGC GGA GGT CCA RCT GCA ACA RTC
YarivH-FOR 5	GTT ATT GCT AGC GGC TCA GCC GGC AAT GGC GCA GGT YCA GCT BCA GCA RTC
YarivH-FOR 6	GTT ATT GCT AGC GGC TCA GCC GGC AAT GGC GCA GGT YCA RCT GCA GCA GTC
YarivH-FOR 7	GTT ATT GCT AGC GGC TCA GCC GGC AAT GGC GCA GGT CCA CGT GAA GCA GTC
YarivH-FOR 8	GTT ATT GCT AGC GGC TCA GCC GGC AAT GGC GGA GGT GAA SST GGT GGA ATC
YarivH-FOR 9	GTT ATT GCT AGC GGC TCA GCC GGC AAT GGC GGA VGT GAW GYT GGT GGA GTC
YarivH-FOR 10	GTT ATT GCT AGC GGC TCA GCC GGC AAT GGC GGA GGT GCA GSK GGT GGA GTC
YarivH-FOR 11	GTT ATT GCT AGC GGC TCA GCC GGC AAT GGC GGA KGT GCA MCT GGT GGA GTC
YarivH-FOR 12	GTT ATT GCT AGC GGC TCA GCC GGC AAT GGC GGA GGT GAA GCT GAT GGA RTC
YarivH-FOR 13	GTT ATT GCT AGC GGC TCA GCC GGC AAT GGC GGA GGT GCA RCT TGT TGA GTC
YarivH-FOR 14	GTT ATT GCT AGC GGC TCA GCC GGC AAT GGC GGA RGT RAA GCT TCT CGA GTC
YarivH-FOR 15	GTT ATT GCT AGC GGC TCA GCC GGC AAT GGC GGA AGT GAA RST TGA GGA GTC

Table B.1: 5' V_H primer mix

YarivH-FOR 16	GTT ATT GCT AGC GGC TCA GCC GGC AAT GGC GCA GGT TAC TCT RAA AGW GTS TG
YarivH-FOR 17	GTT ATT GCT AGC GGC TCA GCC GGC AAT GGC GCA GGT CCA ACT VCA GCA RCC
YarivH-FOR 18	GTT ATT GCT AGC GGC TCA GCC GGC AAT GGC GGA TGT GAA CTT GGA AGT GTC
YarivH-FOR 19	GTT ATT GCT AGC GGC TCA GCC GGC AAT GGC GGA GGT GAA GGT CAT CGA GTC

Table B.1 continued

Primer Name	Primer Sequence
YarivH-REV 1 ApaI	CGA TGG GCCC TTG AAG CTT GCT GAG GAA ACG GTG ACC GTG GT
YarivH-REV 2 ApaI	CGA TGG GCCC TTG AAG CTT GCT GAG GAG ACT GTG AGA GTG GT
YarivH-REV 3 ApaI	CGA TGG GCCC TTG AAG CTT GCT GCA GAG ACA GTG ACC AGA GT
YarivH-REV 4 ApaI	CGA TGG GCCC TTG AAG CTT GCT GAG GAG ACG GTG ACT GAG GT

Table B.2: 3' V_H primer mix

Titer	Mouse 5	Mouse 8	Mouse 9	Mouse 13	Mouse 23
3125000	0.0615	0.054	0.0505	0.051	0.058
625000	0.0685	0.068	0.0535	0.0785	**0.093**
125000	0.068	0.0825	0.061	**0.151**	0.19
25000	**0.119**	**0.1435**	**0.083**	0.477	0.508
5000	0.2805	0.366	0.153	1.28	1.111
1000	1.2445	1.534	0.507	2.7955	3.042

Table B.3: Results of serum titer analysis.

Values represent ELISA absorbance at 450 nm. Values highlighted in red represent the estimated serum titer for the mouse.

Pairwise Tissue Comparison	P-Value comparing true diversity measurements	
	First order (q = 0)	First order (q = 1)
BMPC vs LNPC	0.22	**0.05**
BMPC vs SPPC	0.69	1
LNPC vs SPPC	0.22	**0.02**

Table B.4: Mann-Whitney rank test of diversity indexes

The table shows the p-values of a Mann-Whitney rank test. The test compared diversity between two tissues for all five mice. Values highlighted in red represent tissues whose diversity was statistically significant.

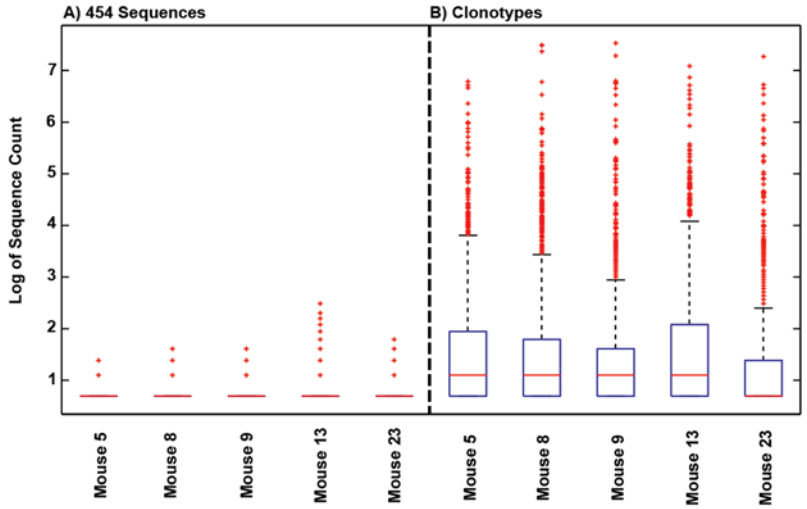


Figure B.1: Distribution of VH gene frequency

A) Box and whisker plot of the number of times we observed a unique V_H gene sequence (at the nucleotide level). B) Box and whisker plot after unique sequence reads were clustered into clonotypes by their respective CDRH3

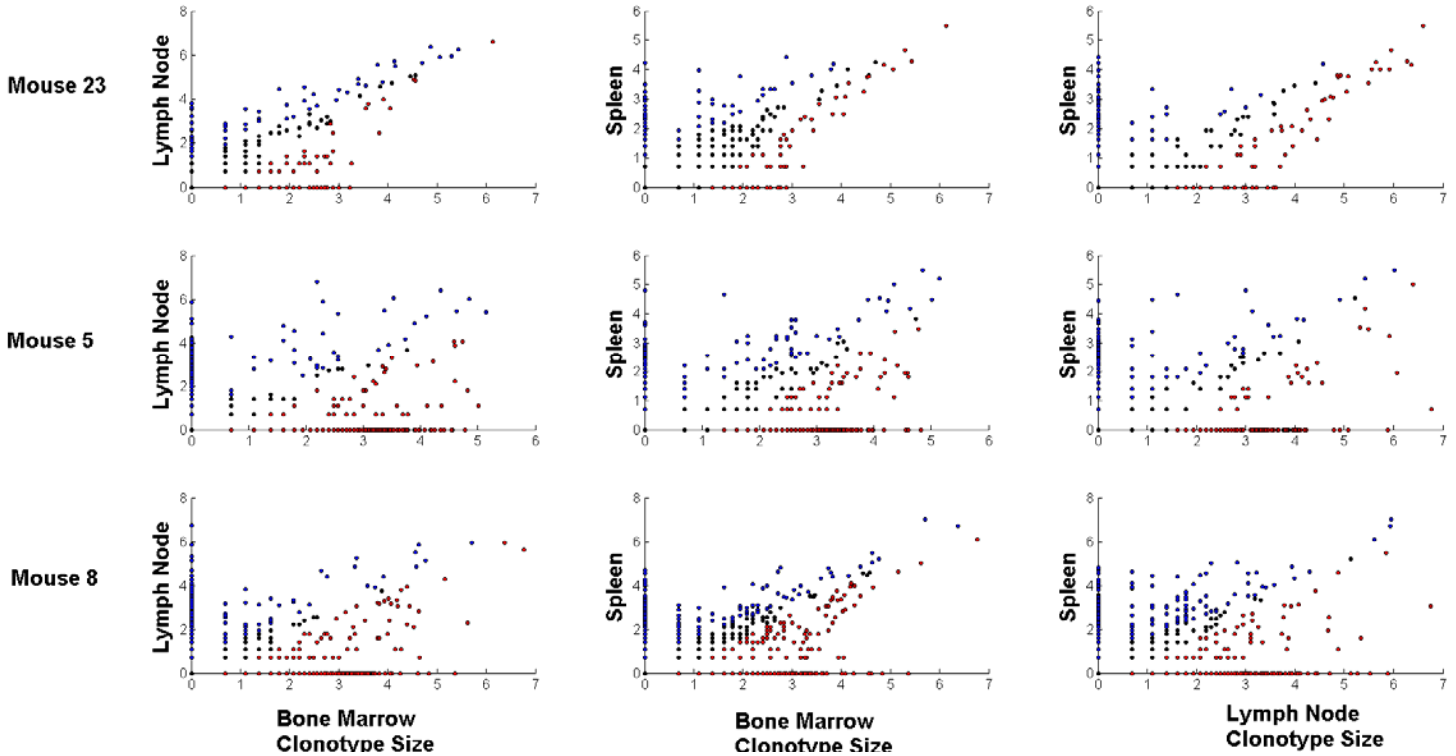


Figure B.2: Scatter plots of clonotype-cluster frequency in pairwise tissues: Mice 5,8, and 23

Scatter plot illustrating the correlation of clonotype-cluster frequency between pairwise tissues. Column 1 plots clonotype-cluster frequency in LNPCvsBMPC; column 2 plots clonotype-cluster frequency in SPPCvsBMPC; column 3 plots clonotype-cluster frequency in SPPC vsLNPC. Top row, middle row, and bottom row show data for mice 23, 5, and 8, respectively

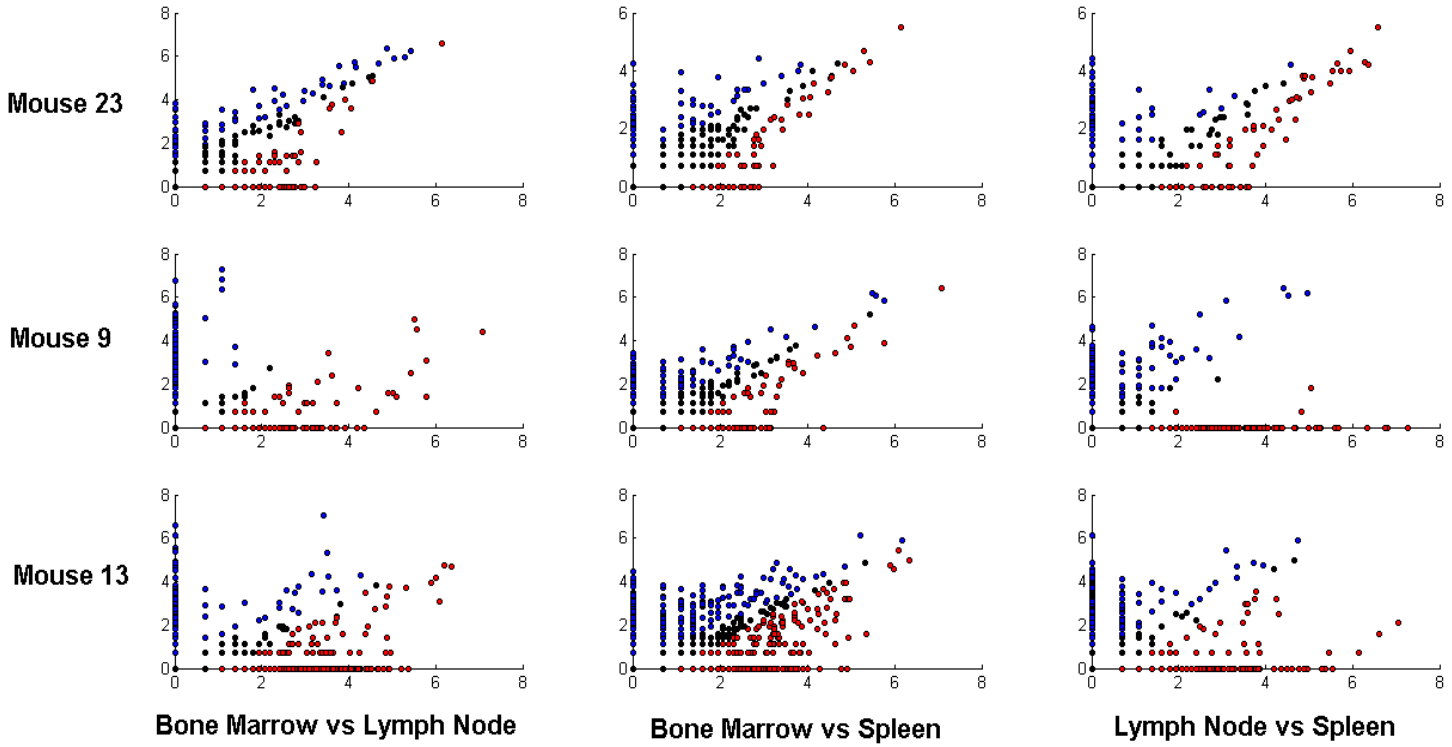


Figure B.3: Scatter plots of clonotype-cluster count in pairwise tissues: Mice 5,8, and 23

Scatter plot illustrating the correlation of clonotype-cluster frequency between pairwise tissues. Column 1 plots clonotype-cluster frequency in LNPCvsBMPC; column 2 plots clonotype-cluster frequency in SPPCvsBMPC; column 3 plots clonotype-cluster frequency in SPPC vsLNPC. Top row, middle row, and bottom row show data for mice 23, 9, and 13, respectively

V_H Gene Variant	Amino Acid Sequence
1_1	QVQLQQSGPELVKPGALVKISCKASGYTFTNYDINWVKQRPGQGLEWIGWINPGDGTTKYSEKFKGKATLTADKSSSTAYMQLSSLTSESSGVFC**AREYGGRGFDY**WGQGTTLTVSS
1_2	QVQLQQSGPDLVKPGALVKISCKASGYTFTSYDINWVKQRPGQGLEWIGWIYPGDGSTKYNEKFKGKATLTADKSSSTAYMQLSSLTSENSAVYFC**AREYGGRGFDY**GGQGTTLTVSS
1_3	EVQLQQSGPELVKPGALVKISCKAFGYTFTSYDINWVKQRPGQGLEWIGWIYPGDGSSKYNEKFKGKATLTADKSSSTAYMQLSSLTSENSAVYFC**AREYGGRGFDY**GGQGTTLTVSS
1_4	QVQLKQSGPELVKPGALVKISCKASGYTFTDYDINWVKQRPGQGLEWIGWINPGDGSAKYSEKFKAKATLTADKSSSTAYMQLSSLTSESSGVYFC**AREYGGRGFDY**WGQGTTLTVSS
1_5	QVQLQQSGPELVKPGALVKISCKASGYTFTSYDINWVKQRPGQGLEWIGWINPGDGSTRYNEKFKGKATVTADKSSSTAYIQLSSLTSENSAVYFC**AREYGGRGFDY**WGQGTTLTVSS
2_1	EVQLQQSGAELMKPGASVIKSCKATGYTFSNYWIGWVKQRPGHGLEWIGEILPGSGSTNYNEKFKGKATFTADSSNATYMQLSSLTEDSAVYYC**ARDSSGGFAY**WGQGTLVTVSA
2_2	EVQLQQSGAELMKPGASVKISCKATGYTFTNYWIGWVKQRPGHGLEWIGEILPGSGRINYNEKFKGKATFTADTSSNATYMQLSSLTSEDSAVYNC**ARDSSGGFAY**WGQGTLVTVSA
3_1	QVQLKQSGAELMKPGASVKISCKATGYTFSSYWIEWVKQRPGHGLEWIGEILPGSGITNYNEKFKGKATFTADTSSNTAYMQLSSLTSEDSAVVYC**ARGGYEGY**WGQGTTLTVSS
3_2	QVQLQQSGAELMKPGASVKISCKATGYTFSSYWIEWVKQRPGHGLEWIGEILPGSGSTNYNEKFKGKATFTADTSSNTAYMQLSSLTSEDSAVVYC**ARGGYEGY**WGQGTTLTVSS
3_3	EVQLQQSGAELMKPGASVKISCKATGYTFSSYWIEWVKQRPGHGLEWIGEILPGSNITNYNEKFKGKATFTADTSSNTAYMQLSSLTSEDSAVYYC**ARGGYEGY**WGQGTTLTVSS
3_4	EVKLVESGAELMKPGASVKISCMATGYTFSGYWMEWVKQRPGHGLEWIGEILPGITNYNEKFKGKATFTADTSSNTAYMQLSSLKTSDDSAVVYC**ARGGYEGY**WGQGTTLTVSS
4_1	EVQLQQSGAELVRPGVSVKISCKGSGYTFTDYAMHWVKQSHAKSLEWIGVISTYYGDVTYNQKFKGKATMTTVDKSSSTAYMELARLTSEDSAIYYC**ARYGNYEGYAMDY**WGQGTSVTVSS
5_1	EVQLVESGPGLVKPSQSLSLTCSVTGYSITSDYYWNWIRQFPNKLEWMGYISYDGYNNYNPSLKNRISITRDTSKNQFFLKLDSVTTEDTATYYC**AKGPYDYFAY**WGQGTLVTSA
5_2	EVKLMESGPGLVKPSQSLSLTCSVTGYSITSDYYWNWLRQFPNKLEWMGYISYDGSNNYNPSLKNRISTRDTSKNQFFLKLNSVTTEDTATYYC**AKGPYDYFAY**WGQGTLVTVSA
5_3	EVKLWESGPGLVKPSQSLSLTCSVTGYSITSGYYWNWIRQFPGNKLEWMGYISYDGSNNYNPSLKNRISITRDTSKNQFFLKLNSVTTEDTATYYC**AKGPYDYFAY**WGQGTLVTVSA

Table B.5: Amino acid sequence of synthesized genes

We tested the top five most predominant clonotype-clusters in the bone marrow, lymph node, and spleen of mouse 23 for HEL-specificity. The following lists the amino acid sequences of the genes we synthesized from the top five clonotype-clusters. The CDRH3 of each clonotype-cluster is highlighted in bold. Specifically we synthesized 5 variants, 2 variants, 4 variants, 1 variant, and 3 variants of the top 5 ranked clonotype-clusters, respectively

Appendix C

THE DERIVATION OF THE CROSS-CORRELATION THEOREM[125]:

1) Let S(i) represent a query DNA sequence as a series of complex numbers where

   A = 0+1j, T = 0-1j, C = 1+0j, and G = 1-0j
2) Let G(i) represent a target DNA sequence as a series of complex numbers

   A = 0+1j, T = 0-1j, C = 1+0j, and G = 1-0j
3) Let F[S(i)] represent the discrete Fourier series of the query DNA sequence

$$S'(\mathrm{t}) = F[S(\mathrm{i})] = \sum_{i=0}^{i=N-1} S(i)e^{-2\Pi jit/N} \quad [\text{Eq C.1}]$$

   Where:

   $S'(\mathrm{t})$ represents the Fourier transform of S(i)

   t represents frequency

   i represents nucleotide position

   I represents the imaginary
4) Similarly, let G'(t) represent the discrete Fourier series (F[G(i)]) of the target DNA sequence, and G'*(t) represent the discrete Fourier series of the complex conjugate of the target DNA Sequence (F[G*(i)])

$$G'(\mathrm{t}) = F[\mathrm{G(i)}](\mathrm{t}) = \sum_{i=0}^{i=N-1} \mathrm{G}(i)e^{-2\Pi jit/N} \quad [\text{Eq C.2}]$$

   Where

   $G'(t)$ represents the Fourier transform of G
5) Thus the Fourier transform of the cross correlation can be derived as follows:

$$s\Diamond g(d) = \sum_{i=0}^{i=N-1} \mathrm{S}(i) \bullet G^*(i-d)$$

$$F[s\Diamond g(d)](\mathrm{t}) = \sum_{d=0}^{d=N-1} [\sum_{i=0}^{i=N-1} \mathrm{S}(i) \bullet G^*(i-d)] e^{-2\Pi j dt/N}$$

$$F[s\Diamond g(d)](\mathrm{t}) = \sum_{i=0}^{i=N-1} \mathrm{S}(i) \bullet [\sum_{d=0}^{d=N-1} \mathrm{G}^*(i-d) e^{-2\Pi j dt/N}]$$

Let y = d-i, then

$$F[s\Diamond g(d)](\mathrm{t}) = \sum_{i=0}^{i=N-1} \mathrm{S}(i) \bullet [\sum_{y=0}^{y=N-1} \mathrm{G}^*(-\mathrm{y}) e^{-2\Pi j(\mathrm{y}+\mathrm{i})t/N}]$$

$$F[s\Diamond g(d)](\mathrm{t}) = \sum_{i=0}^{i=N-1} \mathrm{S}(i) \bullet [\sum_{y=0}^{y=N-1} \mathrm{G}^*(-y) e^{-(2\Pi jit + 2\Pi jty)/N}]$$

$$F[s\Diamond g(d)](\mathrm{t}) = \sum_{i=0}^{i=N-1} \mathrm{S}(i) \bullet [\sum_{y=0}^{y=N-1} \mathrm{G}^*(-y) e^{-2\Pi jyt/\mathrm{N}}] e^{-2\Pi jti/N}$$

$$F[s\Diamond g(d)](\mathrm{t}) = \sum_{i=0}^{i=N-1} \mathrm{S}(i) e^{-2\Pi jti/N} \bullet F[G^*(-y)]$$

$$F[s\Diamond g(d)](\mathrm{t}) = F[\mathrm{S(i)}] \bullet Conj(G'(y)) = \mathrm{S'(t)} \bullet Conj(G'(\mathrm{t}))$$ [Eq C.3]

THRESHOLD SCORE FOR GERMLINE ASSIGNMENT ALGORITHM

An important step in our germline assignment algorithm is the identification of alignment scores that result from non-random nucleotide alignment of the sequence of interest to each germline. Therefore, we need a threshold score that conveys regions of non-random alignment between the sequence and germline. We can model the expected number of matching base-pairs in a nucleotide alignment between random sequences using the binomial distribution. The binomial distribution models the probability of observing k successes in N trials given a probability of success, p:

$$\Pr(X = k) = \binom{n}{k} p^k (1-p)^{(n-k)} \quad \text{[Eq C.4]}$$

Where
Pr(X=k) = Probability of observing k successes
n = number of trials
k = number of successes
p = probability of success

With respect to nucleotide alignment, a "success" can be thought of as matching base pairs and the number of trials can be thought of as the length of the nucleotide alignment. More importantly, the expected number of success in N trials can be modeled as follows:

$$E[X] = np \quad \text{[Eq C.4]}$$
$$Var[X] = np(1-p) \quad \text{[Eq C.5]}$$

Where
E[X] = expected number of successes in n trials
Var[X] = variance in the expected number of successes
n= number of trials
p = frequency of success

Assuming that nucleotide alignment is completely random, then we would expect the probability of "success" is:

$$p = P(A)\,\mathrm{P}(A) + P(C)P(C) + \mathrm{P}(\mathrm{T})\,\mathrm{P}(\mathrm{T}) + P(G)\,\mathrm{P}(G)$$
$$p = 4 * (0.25 * 0.25)$$
$$p = 0.25$$

Where

p = probability of matching nucleotide pair

P(nt) = Probablity of seeing a specific base = 0.25 (1/4)

Therefore, using these equations, the expected number of random base-pair matches is:

$$E[\text{match}] = \mathrm{E}[\text{comp\_mismatch}] = np$$
$$E[match] = \mathrm{E}[\text{comp\_mismatch}] = 0.25 * n$$
$$\text{Var} = 0.25 * n(1 - 0.25)$$

Where

n = length of alignment

E[match]=Expected number of matching nucleotides

E[comp_mismatch] =Expected number of A-T/C-G mismatches

Var=variance in the expected matches

In our alignment algorithm, the FFT function scores matching nucleotides with a +1 score and penalizes mismatching complementary nucleotides by -1. Therefore, the alignment score will just be the sum of two random variables (matching nucleotides) and (mismatching nucleotides). In this case, the expectation of the sum of the two random variables will be 0. However, we are most interested in the variance of the alignment score. The variance of two randomly distributed binomial values is as follows:

Exp[Sc]=(m*E[match]+mm*E[comp_mismatch])

The variance of the score can be modeled as the variance of the sum of two random variables (x,y):

Var[a*x+b*y]=$a^2 * \mathrm{var}(x) + b^2 * \mathrm{var}(y) + 2 * a * b * \mathrm{Cov}(x, y)$

Var[Sc]=Var(m*E[match]+mm*E[comp_mismatch])

Var[Sc] = (m^2Var[match]+mm^2Var[comp_mismatch]+2*m*mm*Cov(match,comp_mismatch))

Let

Cov(match,comp_mismatch)= n*(p(match and comp_mismatch)-p(match)*p(comp_mismatch))

m=1

mm=-1

p(match and comp_mismatch) = 0

Then

Var[Sc] = $n * (0.25) * (1 - 0.25)$+n*(0.25)*(1-0.25)+(n*2*0.25^2) [Eq C.5]

Equation C.5 models the expected variance in alignment score as a function of the alignment length. For the average alignment length of 300 base pairs (average length of a germline) we expect the standard deviation in the alignment score using equation C.5 to be approximately 12. Therefore, we use Equation C.5 to select our threshold for non-random nucleotide alignments as a function of the alignment length. In the current algorithm, this threshold is set at 3 standard deviations above the expected random alignment score.

EXAMPLES OF COMPLEX SERIES REPRESENTATIONS OF DNA SEQUENCES

Before we use the FFT to align nucleotide sequences, we first have to convert the nucleotide sequence into a series of complex integers. In this study, we represented DNA as follows:

Let
A=0+1j
T=0-1j
C=1+0j
G=-1+0j

Performing the cross-correlation function on DNA sequences represented as this series resulted in the following match and mismatch scores between nucleotides:

	A	C	T	G
A	(0+1j)*(0-1j)	(0+1j)*(1-0j)	(0+1j)*(0+1j)	(0+1j)*(-1-0j)
C	(1+0j)*(0-1j)	(1+0j)*(1+0j)	(1+0j)*(0+1j)	(1+0j)*(-1-0j)
T	(0-1j)*(0+1j)	(0-1j)*(1-0j)	(0-1j)*(0+1j)	(0-1j)*(-1-0j)
G	(-1+0j)*(0-1j)	(-1+0j)*(1-0j)	(-1+0j)*(0+1j)	(-1+0j)*(-1+0j)

Calculation of Alignment Scores

	A	C	T	G
A	1	0	-1	0
C	0	1	0	-1
T	-1	0	1	0
G	0	-1	0	1

Table of Alignment Scores

Using this representation and excluding imaginary coefficients, all nucleotide matches were given a +1 score whereas complimentary (A-T and C-G) mismatches were given a penalty of -1. All other possible combinations such as A-C and T-G were not penalized (score = 0). Although we use the method above for our algorithm, it is not the only allowed

nucleotide transformation for performing nucleotide alignments. For example, the following demonstrates how we can design a scoring system that only scores matching nucleotides:

Let
A=[0 + 1j, 1+ 0j]
T=[0 - 1j, 1 + 0j]
C=[1 + 0j, 0 +1j]
G=[-1+ 0j, 0 + 1j]

In this example, we represent nucleotides as a 2-D vector, or as two different complex series. If we perform the cross-correlation on each vector and then subsequently take the sum of the real coefficients we find that all base-pair matches are scored by +2 whereas all other combinations are neither scored nor penalized.

	A	C	T	G
A	Re[(0+1j)*(0-1j)]+ Re[(1+0j)*(1-0j)]	Re[(0+1j)*(1+0j)]+ Re[(1+0j)*(0+1j)]	Re[(0+1j)*(0+1j)]+ Re[(1+0j)*(1+0j)]	Re[(0+1j)*(-1+0j)]+ Re[(1+0j)*(0+1j)]
C	Re[(1+0j)*(0+1j)]+ Re[(0+1j)*(1+0j)]	Re[(1+0j)*(1+0j)]+ Re[(0+1j)*(0-1j)]	Re[(1+0j)*(0+1j)]+ Re[(0+1j)*(1+0j)]	Re[(1+0j)*(-1+0j)]+ Re[(0+1j)*(0-1j)]
T	Re[(0-1j)*(0-1j)]+ Re[(1+0j)*(1+0j)]	Re[(0-1j)*(1+0j)]+ Re[(1+0j)*(0+1j)]	Re[(0-1j)*(0+1j)]+ Re[(1+0j)*(1+0j)]	Re[(0-1j)*(-1-0j)]+ Re[(1+0j)*(0+1j)]
G	Re[(-1+0j)*(0-1j)]+ Re[(0+1j)*(1+0j)]	Re[(-1+0j)*(1+0j)]+ Re[(0+1j)*(0-1j)]	Re[(-1+0j)*(0+1j)]+ Re[(0+1j)*(1+0j)]	Re[(-1+0j)*(-1+0j)]+ Re[(0+1j)*(0-1j)]

Calculation of Alignment Scores

	A	C	T	G
A	2	0	0	0
C	0	2	0	0
T	0	0	2	0
G	0	0	0	2

Table of Alignment Scores

Finally, we show that representing nucleotides as a 2-D matrix with 5 separate complex series functions allows one to define any score for matching nucleotides and any score for mismatching nucleotides.

Let
m=Match Score
q=Mismatch Score
$A=[(m+q)^{1/2},0,0,0,q^{1/2}]$
$T=[0,(m+q)^{1/2},0,0,\ q^{1/2}]$
$C=[0,0,(m+q)^{1/2},0,\ q^{1/2}]$
$G=[0,0,0,(m+q)^{1/2},\ q^{1/2}]$

In this example, each nucleotide represents an individual series of complex numbers (columns 1-4). Moreover, the last complex series (column 5) represents the "mismatch score" column. That is, the last column will always refer to the mismatch value we give to base-pair mismatches. Given these values for A,C,T, and G, we can calculate the alignment between aligning adenine to adenine (A-A) and aligning adenine to thymine (A-T) function as follows:

A-A Alignment:

$$[(m+q)^{1/2}*(m+q)^{1/2}]+[0*0]+[0*0]+[0*0]-[q^{1/2}*q^{1/2}]$$

$(m+q)-q = m$

A-T Alignment:

$$[(m+q)^{1/2}*0]+[0*(m+q)^{1/2}]+[0*0]+[0*0]-[q^{1/2}*q^{1/2}]$$

$-q = -q$

The key difference in this example, is that we always subtract the cross-correlation calculated in the last column (column 5). Repeating the steps above for all other possible

combinations we show that we the alignment score matrix using the series above will be as follows:

	A	C	T	G
A	m	-q	-q	-q
C	-q	m	-q	-q
T	-q	-q	m	-q
G	-q	-q	-q	m

Table of Alignment Scores

The purpose of these three examples is to illustrate that there are many different methods for scoring nucleotide alignments using the cross-correlation function, and thus the use of the FFT for nucleotide alignment.

EXAMPLE OF NUCLEOTIDE ALIGNMENT USING SPARSE FFT

The following is a simple illustration of how a sparse FFT could be used to identify the maximum alignment diagonal between to sequences. I would like to acknowledge Haitham Hassanieh (Dr. Katabi lab, M.I.T) for his assistance in investigating the application of SparseFFT for nucleotide alignments

```
S = seqConvert(seq,3); %convert sequence into a series of complex
integers=> A=1, T = -1, C = 1i, G = -1i
S(880,1) = 0; %pad the complex series with zeros;
seqFFT = fft(seqConvert(seq,3)); %transform the sequence into its
fourier series using FFTW3
g = seqConvert(germline,3); %convert the germline VH sequence into a
series of complex integers
g(880,1) = 0; %pad the complex series with zeros;
clusterFFT = fft(g); %transform the sequence into its fourier series
using FFTW3
algn = seqFFT.*conj(clusterFFT);% perform cross correlation in fourier

%%Take the FULL inverse FFT of the cross correlation
alignedSeq1 = (real(ifft(algn)));
[Val,Shift1] = max(alignedSeq1);
%%%%%%
%%The following is an example of a sparseFFT method
%first sparse alignment.  Take the inverse of the subsampled “algn”
vector.  Subsample every 16 coefficients.
Filter1 = real(ifft(algn(1:16:end)));
 %second spares alignment.  Take another inverse of a new subsampled
“algn” vector.  Subsample every 11 coefficients this time
Filter2 = real(ifft(algn(1:11:end)));
[V1, I1] = max(Filter1);
[V2, I2] = max(Filter2);
%The actual position of the maximum alignment is predicted by noting
the only coefficient that was subsampled in both filters.  This
equation identifies that coefficient.
Shift2 = mod(mod(I1,55)*16*31+mod(I2,16)*55*7,880);
%%%%%%
[Shift1; Shift2] %shift1 should be equal to shift 2 if done correctly
```

SUPPLEMENTARY INFORMATION

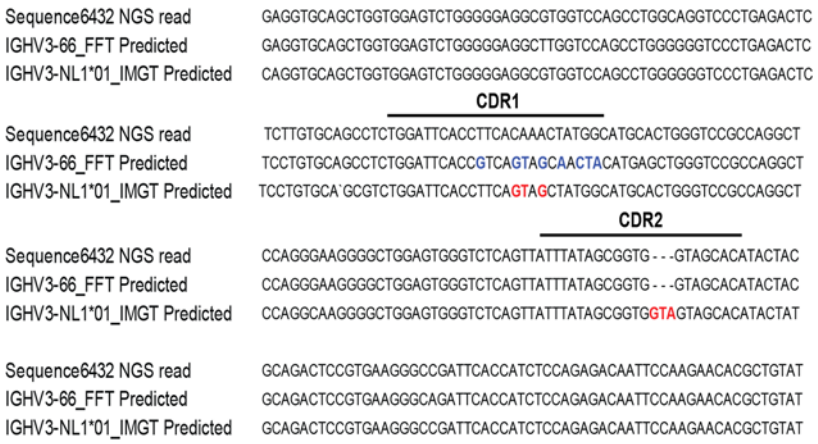


Figure C.1: Alignment of NGS read to germline IGHV3-NL1 and IGHV3-66

In this example, Smith-Waterman alignment shows that the alignment to IGHV3-66 is better than the alignment to IGHV3-NL1 because it does not result in the addition of in-del mutations in the CDR2 region. However, aligning the sequence to IGH3-66 results in five additional bp mismatch errors in the CDR1 region as compared to IGHV3-NL1 alignment

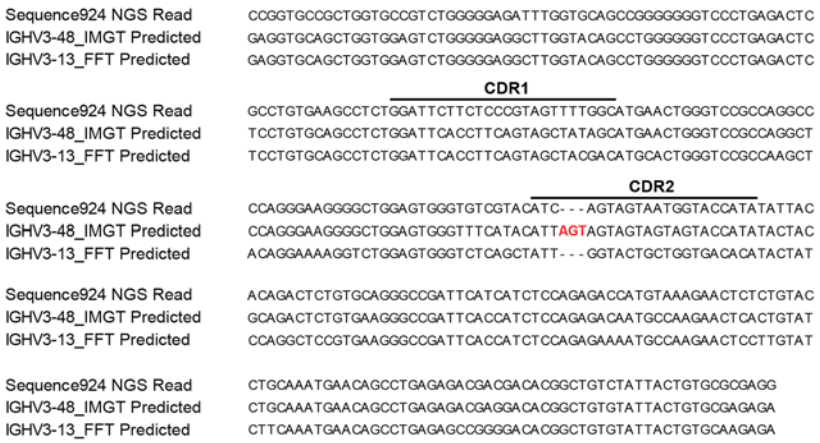


Figure C.2: Alignment of NGS read to germline IGHV3-48 and IGHV3-13

The FFT algorithm cannot account for gaps and consequently predicts that IGHV3-13 is a better alignment because there are no in-del mutations between the NGS read and IGHV3-13.

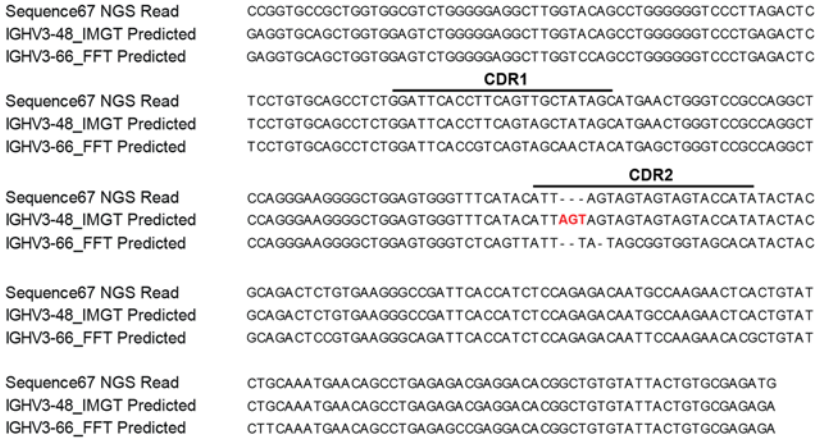


Figure C.3: Alignment of NGS read to germline IGHV3-48 and IGHV3-66

The FFT algorithm cannot account for gaps and consequently predicts that IGHV3-66 is a better alignment because there are no in-del mutations between the NGS read and IGHV3-66.

Cluster	# Clustered V_H genes	Clustered V_H family	Consensus Sequence of Cluster
1	8	IGHV5	GAGGTGCAGCTGGTGCAGTCTGGAGCAGAGGTGAAAAAGCCCGGGGAGTCTCTGAGGATCTCCTGTAAGGGTTCTGGATACAGCT TTACCAGCTACTGGATCGGCTGGGTGCGCCAGATGCCCGGGAAAGGCCTGGAGTGGATGGGGAGCATCGATCCTGGTGACTCTGA TACCAGCTACAGCCCGTCCTTCCAAGGCCAGGTCACCATCTCAGCCGACAAGTCCATCAGCACCGCCTACCTGCAGTGGAGCAGCC TGAAGGCCTCGGACACCGCCCCGTATTACTGTGCGCGACA
2	1	IGHV5	GAAGTGCAGCTGGTGCAGTCTGGAGCAGAGGTGAAAAAGCCCGGGGAGTCTCTGAGGATCTCCTGTAAGGGTTCTGGATACAGCT TTACCAGCTACTGGATCAGCTGGGTGCGCCAGATGCCCGGGAAAGGCTTGGAGTGGATGGGGAGGATTGATCCTAGTGACTCTTAT ACCAACTACAGCCCGTCCTTCCAAGGCCACGTCACCATCTCAGCTGACAAGTCCATCAGCACTGCCTACCTGCAGTGGAGCAGCCT GAAGGCTCGGACACCGCCATGTATTACTGTGCGAGACA
3	66	IGHV3	CAGGTGCAGCTGGTGGAGTCTGGGGGAGGCGTGGTCCAGCCTGGGGGGGTCCCTGAGACTCTCCTGTGCAGCCTCTGGATTCACCTT CAGTAGCTATGCCATGCACTGGGTCCGCCAGGCTCCAGGCAAGGGGCTGGAGTGGGTGGCAGTTATTAGTTGTGATGGAAGTAAC ACATACTACGCAGACTCCGTGAAGGGCCGATTCACCATCTCCAGAGACAACTCCAAGAACACGCTGTATCTGCAAATGAACAGCC TGAGAGCCGAGGACACGGCTGTGTATTACTGTGCGAGAGATA
4	31	IGHV1	CAGGTCCAGCTGGTGCAGTCTGGGGCTGAGGTGAAGAAGCCTGGGGCCTCGGTGAAGGTCTCCTGCAAGGCTTCTGGAGGCACCT TCACCAGCTATGCTATCCGCTGGGTGCGACAGGCCCCTGGACAAGGGCTTGAGTGGATGGGAGGGATCATCCCTATCATTGGTACC GCAAACTACGCACAGAAGTTCCAGGGCAGAGTCACCATGACCGCGGACACATCCACGAGCACAGCCTACATGGAGCTGAGCAGCC TGAGATCTGAGGACACGGCCGTGTATTACTGTGCGAGAGA
5	13	IGHV3	GAGGTGCAGCTGGTGGAGTCTGGGGGAGGCTTGGTCCAGCCTGGGGGGTCCCTGAGACTCTCCTGTGCAGCCTCTGGATTCACCGT CAGTAGCAACTACATGCGCTGGGTCCGCCAGGCTCCAGGGAAGGGGCTGGAGTGGGTCTCAGCTATTTATAGCGGTGGTAGCACA TACTACGCAGACTCCGTGAAGGGCCGATTCACCATCTCCAGAGACAATTCCAAGAACACGCTGTATCTTCAAATGAACAGCCTGAG AGCCGAGGACACGGCCGTGTATTACTGTGCGAGAGA
6	4	IGHV3	GAGGTGCAGCTGGTGGAGTCCGGGGGAGGCTTGGTCCAGCCTGGGGGGTCCCTGAAACTCTCCTGTGCAGCCTCTGGGTTCACCTT CAGTGGCCCCGCCATGCACTGGGTCCGCCAGGCTCCCGGGAAGGGGCTGGAGTGGGTTGGCCGTACTAGAAGCAAAGCTAACAGT TACGCCACAGCATACGCCGCGTCGGTGAAAGGCAGGTTCACCATCTCCAGAGATGATTCAAAGAACACGCCGTATCTGCAAATGA ACAGCCTGAAAACCGAGGACACGGCCGTGTATTACTGTACTAGACA
7	37	IGHV4	CAGGTGCAGCTGCAGGAGTCGGGCCCAGGACTGGTGAAGCCTTCGGAGACCCTGTCCCTCACCTGCGCTGTCTCTGGTGGCTCCAT CAGCAGCGGTGGTTACTACTGGAGCTGGATCCGCCAGCCCCCAGGGAAGGGCCTGGAGTGGATTGGGTACATCTATCACAGTGGG AGCACCTACTACAACCCGTCCCTCAAGAGTCGAGTCACCATATCAGTAGACACGTCCAAGAACCAGTTCTCCCTGAAGCTGAGCTC TGTGACCGCCGCGGACACGGCCGTGTATTACTGTGCGAGAGA
8	17	IGHV4	CAGGTGCAGCTGCAGGAGTCGGGCCCAGGACTGGTGAAGCCTTCGGAGACCCTGTCCCTCACCTGCGCTGTCTCTGGTTACTCCAT CAGCAGTAGTAACTGGTGGGGCTGGATCCGGCAGCCCCCAGGGAAGGGGCTGGAGTGGATTGGGGACATCTATCATAGTGGGAGC ACCTACTACAACCCGTCCCTCAAGAGTCGAGTCACCATGTCAGTAGACACGTCCAAGAACCAGTTCTCCCTGAAGCTGAGCTCTGT GACCGCCGCGGACACGGCCGTGTATTACTGTGCGAGAGA
9	2	IGHV4	CAGGTGCAGCTGCAGGAGTCGGGCCCAGGACTGGTGAAGCCTTCGCAGACCCTGTCCCTCACCTGCACTGTCTCTGGTGGCTCCGT CAGCAGTGGTGGTTACTACTGGAGCTGGATCCGCCAGCCCCCAGGGAAGGGCCTGGAGTGGATTGGGTACATCTATTACAGTGGG AGCACCAACTACAACCCCTCCCTCAAGAGTCGAGTCACCATATCAGTAGACACGTCCAAGAACCAGTTCTCCCTGAAGCTGAGCTC TGTGACCGCGGACGCGGCCGTGTATTACTGTGCG

Table C.1: Table of V_H clusters used in germline assignment algorithm

10	23	IGHV4	CAGGTGCAGCTGCAGCAGTCGGGCCCAGGACTGGTGAAGCCTTCGGAGACCCTGTCCCTCACCTGCGCTGTCTCTGGTGGCTCCTT CAGTGGTTACTACTGGAGCTGGATCCGCCAGCCCCCAGGGAAGGGGCTGGAGTGGATTGGGGAAATCTATCACAGTGGGAGCACC AACTACAACCCGTCCCTCAAGAGTCGAGTCACCATATCAGTAGACACGTCCAAGAACCAGTTCTCCCTGAAGCTGAGCTCTGTGAC CGCCGCGGACACGGCCGTGTATTACTGTGCGAGAGA
11	3	IGHV1	CAGGTCCAGCTGGTACAGTCTGGGGCTGAGGTGAAGAAGCCTGGGGCCACAGTGAAAATCTCCTGCAAGGTTTCCGGATACACCC TCACCGACTACTCCATGCACTGGGTGCAACAGGCCCCTGGAAAAGGGCTTGAGTGGATGGGACTTGTTGATCCTGAAGATGGTGA AACAATCTACGCAGAGAAGTTCCAGGGCAGAGTCACCATAACCGCGGACACGTCTACAGACACAGCCTACATGGAGCTGAGCAGC CTGAGATCTGAGGACACGGCCGTGTATTACTGTGCAACAGA
12	25	IGHV2	CAGGTCACCTTGAAGGAGTCTGGTCCTGCGCTGGTGAAACCCACACAGACCCTCACGCTGACCTGCACCTTCTCTGGGTTCTCACT CAGCACTAGTGGAATGCGTGTGAGCTGGATCCGTCAGCCCCCAGGGAAGGCCCTGGAGTGGCTTGCACTCATTGATTGGGATGAT GATAAGCGCTACAGCCCATCTCTGAAGACCAGGCTCACCATCTCCAAGGACACCTCCAAAAACCAGGTGGTCCTTACAATGACCA ACATGGACCCTGTGGACACAGCCACGTATTACTGTGCACGCAGAC
13	3	IGHV1	CAGATGCAGCTGGTGCAGTCTGGGGCTGAGGTGAAGAAGACTGGGTCCTCAGTGAAGGTTTCCTGCAAGGCTTCCGGATACACCTT CACCTACCGCTACCTGCACTGGGTGCGACAGGCCCCCGGACAAGCGCTTGAGTGGATGGGATGGATCACACCTTTCAATGGTAAC ACCAACTACGCACAGAAATTCCAGGACAGAGTCACCATTACCAGGGACAGGTCTATGAGCACAGCCTACATGGAGCTGAGCAGCC TGAGATCTGAGGACACAGCCATGTATTACTGTGCAAGANA
14	2	IGHV3	GAGGTGCAGCTGGTGGAGTCTCGGGGAGGCTTGGTACAGCCTGGGGGGTCCCTGAGACTCTCCTGTGCAGCCTCTGGATTCACCGT CAGTAGCAATGAGATGAGCTGGGTCCGCCAGGCTCCAGGGAAGGGTCTGGAGTGGGTCTCATCCATTAGTGGTGGTAGCACATAC TACGCAGACTCCAGGAAGGGCAGATTCACCATCTCCAGAGACAATTCCAAGAACACGCTGCATCTTCAAATGAACAGCCTGAGAG CTGAGGGCACGGCCGTGTATTACTGTGCCAGAGAAA
15	5	IGHV7	CAGGTGCAGCTGGTGCAATCTGGGTCTGAGTTGAAGAAGCCTGGGGCCTCAGTGAAGGTTTCCTGCAAGGCTTCTGGATACACCTT CACTAGCTATGCTATGAATTGGGTGCGACAGGCCCCTGGACAAGGGCTTGAGTGGATGGGATGGATCAACACCAACACTGGGAAC CCAACGTATGCCCAGGGCTTCACAGGACGGTTTGTCTTCTCCTTGGACACCTCTGTCAGCACGGCATATCTGCAGATCAGCAGCCT AAAGGCTGAGGACACTGCCGTGTATTACTGTGCGAGAGA
16	8	IGHV3	GAGGTGCAGCTGGTGGAGTCTGGGGGAGGCTTGGTAAAGCCTGGGGGGTCCCTTAGACTCTCCTGTGCAGCCTCTGGATTCACTTT CAGTAACGCCTGGATGAGCTGGGTCCGCCAGGCTCCAGGGAAGGGGCTGGAGTGGGTTGGCCGTATTAAAAGCAAAACTGATGGT GGGACAACAGACTACGCTGCACCCGTGAAAGGCAGATTCACCATCTCAAGAGATGATTCAAAAAACACGCTGTATCTGCAAATGA ACAGCCTGAAAACCGAGGACACAGCCGTGTATTACTGTACCACAGA
17	5	IGHV3	GAGGTGCAGCTGGTGGAGTCTGGGGGAGGCTTGGTACAGCCAGGGCGGTCCCTGAGACTCTCCTGTACAGCTTCTGGATTCACCTT TGGTGATTATGCTATGAGCTGGTTCCGCCAGGCTCCAGGGAAGGGGCTGGAGTGGGTAGGTTTCATTAGAAGCAAAGCTTATGGTG GGACAACAGAATACGCCGCGTCTGTGAAAGGCAGATTCACCATCTCAAGAGATGATTCCAAAAGCATCGCCTATCTGCAAATGAA CAGCCTGAAAACCGAGGACACAGCCGTGTATTACTGTACTAGAGA
18	2	IGHV1	CAAATGCAGCTGGTGCAGTCTGGGCCTGAGGTGAAGAAGCCTGGGACCTCAGTGAAGGTCTCCTGCAAGGCTTCTGGATTCACCTT TACTAGCTCTGCTGTGCAGTGGGTGCGACAGGCTCGTGGACAACGCCTTGAGTGGATAGGATGGATCGTCGTTGGCAGTGGTAACA CAAACTACGCACAGAAGTTCCAGGAAAGAGTCACCATTACCAGGGACATGTCCACAAGCACAGCCTACATGGAGCTGAGCAGCCT GAGATCCGAGGACACGGCCGTGTATTACTGTGCGGCAGA
19	2	IGHV6	CAGGTACAGCTGCAGCAGTCAGGTCCGGGACTGGTGAAGCCCTCGCAGACCCTCTCACTCACCTGTGCCATCTCCGGGGACAGTGT CTCTAGCAACAGTGCTGCTTGGAACTGGATCAGGCAGTCCCCATCGAGAGGCCTTGAGTGGCTGGGAAGGACATACTACAGGTCC AAGTGGTATAATGATTATGCAGTATCTGTGAAAAGTCGAATAACCATCAACCCAGACACATCCAAGAACCAGTTCTCCCTGCAGCT GAACTCTGTGACTCCCGAGGACACGGCTGTGTATTACTGTGCAAGAGA

Table C.1 continued

Appendix D

SUPPLEMENTAL FIGURES AND TABLES

Metadata	Description	Example
Experiment ID	Descriptive ID that describes the sample	Mouse23_IgG_LNPC
Project ID	Project name	MouseTissueProject
Date	Date of sequencing	3/4/2014
Work Order	Work order number received from GSAF	JA12345
NGS Platform	Next Generation Sequencing platform	454
Isotype	Isotype of immunoglobulin sequenced	IgG
Keywords	project description	Mouse23 lymph nodes
Chain Type	Immunoglobulin chain sequenced (Heavy, Light, Alpha, Beta)	HEAVY
Cell Type	Cell types present in sample sequences	PC
Members	List of members with write access to data	Constantine Chrysostomou
Species	Species collected from	Human
Read access	List of members with read access to data	All
Reference	Publication referring to this study	
Pairing technique	Technique used to pair heavy and light chains, if applicable	
Raw data	Filenames of raw fastq sequencing data	
IMGT	Name of the folder containing unzipped IMGT files	IMGT_LNPC
LAB	Name of the research lab	GEORGIOU
Cell markers	List of cell markers used for positive or negative selection	CD138+
Cell number	Attempted number of cells sequenced	

Table D.1: Metadata stored in IRODs

Isotype	Barcode
IgG	AAAAATGGGCCCTGCGATGGGCCCTTGGTGGAGG
IgM	AAAAATGGGCCCTGGGTTGGGGCGGATGCACTCC
IgA	AAAAATGGGCCCTGCTTGGGGCTGGTCGGGGATG
IgK	AAAAGTGCGGCCGCAGATGGTGCAGCCACAGTTC
IgL	AAAAGTGCGGCCGCGAGGGCGGGAACAGAGTGAC

Table D.2: List of barcode sequences for isotype annotation

Length	CDR3	Frequency
5	CARAC	0.1
0		0.05
6	CARAEW	0.01

A) Text Tab Delimited Format

```
5\tCARAC\t0.1
0\t\t0.05
6\tCARAEW\t0.01
```

B) JSON Format

```
{"Length":5","CDR3":"CARAC","Frequency":0.1}
{"Length":0,"Frequency":0.05}
{"Length":6","CDR3":"CARAEW","Frequency":0.01}
```

Figure D.1: Illustration of the JSON file format

The JSON file format (B) relies on "strings" to define fields as compared to the tab-delimited counterpart (A). Each "row" of the JSON file does not have to output information from each field. For example, the CDR3 in row two is undefined, so the JSON file does not need to include it. Also, the CDR3 field can be located anywhere as compared the table where it must always be in the second column.

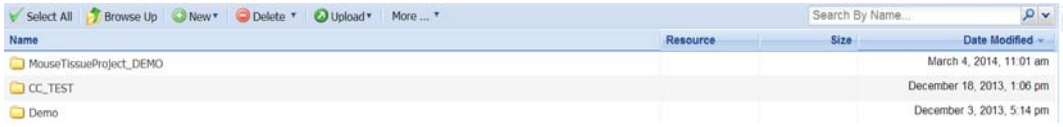
RUNNING IMMUNOGREP ON APPSOMA

The following provides a detailed analysis of running the immunoglobulin repertoire analysis pipeline on APPSOMA. For this analysis we will use 454 data from the data reported in Chapter 2. Specifically, we have three unprocessed files of sequence data for Mouse 23 bone marrow, lymph nodes, and spleen stored in the IRODs system.

Storing raw data stored on IRODs

Our IRODs system stores raw sequencing data compiled from our lab over the last three years. For each experiment we upload the following data to IRODs: (1) fasta file of sequence data (2) fastq files containing sequence quality scores, (3) IMGT analysis for that experiment (if performed). Most importantly, for every experiment file we store metadata that describes the sequencing experiment (Table D.1).

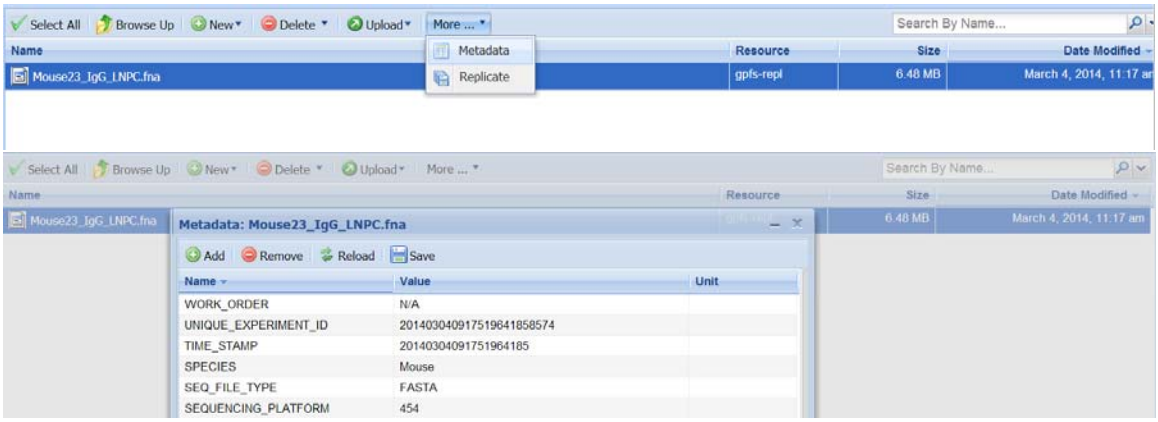
1) The IRODs system comes with an interactive web interface for downloading data. In our file structure framework, each parent folder in the IRODs system is defined by the PROJECT_ID that defines a project. One project can have multiple experiments. Open this parent folder to view subfolders within corresponding to experiments



2) In this case, the project, MouseTissueProject_DEMO contains three separate experiments for each sequencing dataset: Mouse_23_IgG_LNPC, Mouse 23_IgG_SPPC, and Mouse_23_IgG_BMPC corresponding to 454 data for the immunoglobulin data.

3) Opening the experiment folder contains subfolders that store 1) FASTA folder containing the FASTA file, 2) RAW folder that contains the optional raw fastq files with quality scores, and 3) IMGT folder that contains the optional IMGT analysis.

4) The fasta folder only contains one file, corresponding to the quality filtered and optionally flash processed raw data into a FASTA file. Importantly, all metadata (Table D.1) is linked directly to this file.



ImmunoGReP on APPSOMA

Steps for adding an experiment to the HTS Immunoglobulin database using

1) Step 1: Raw data stored on IRODs

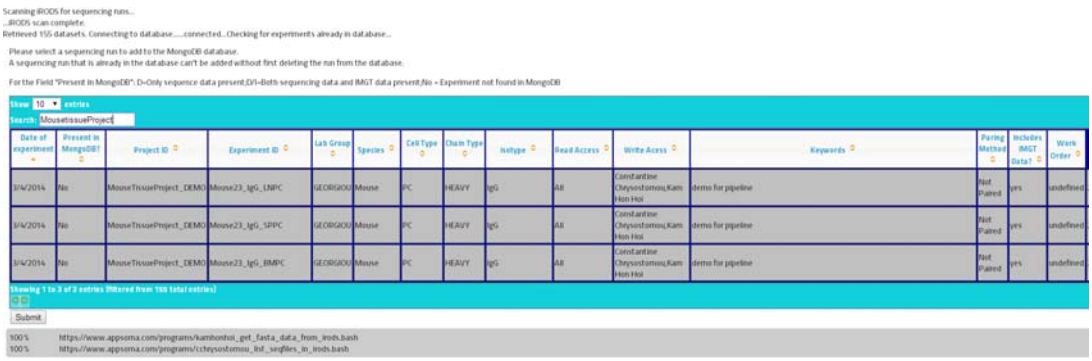
2) Step 2: Go to APPSOMA (https://appsoma.austin.utexas.edu/home)

3) Go to the list of APPS by clicking the APPS tab in the top left corner

4) In the search bar, search for the APP called “Add Experiment From IRODs to database”

5) Open the APP and select the executive node to use for running the script. In this case we will select the node UTEXAS2.

6) The APP will proceed to access the RAW data on IRODs and search for all datasets that you have access too. Information of your experiments are listed in an interactive table Only users with "Write access" can view the raw data files stored on IRODs.



7) Select the experiment of interest, in this case, Mouse23_IgG_LNPC, and press submit

8) The program then proceeds to add the information to the HTS Immunological database. If this file also contains IMGT data from IRODs, which this file does, the IMGT analysis info is automatically added to the database under the IMGT Analysis field. Repeat steps 1-8 for the remaining datasets Mouse23_IgG_BMPC and Mouse23_IgG_SPPC to add all datasets for the experiment to the database

Steps for querying experiments from the HTS immunoglobulin database using ImmunoGReP on APPSOMA

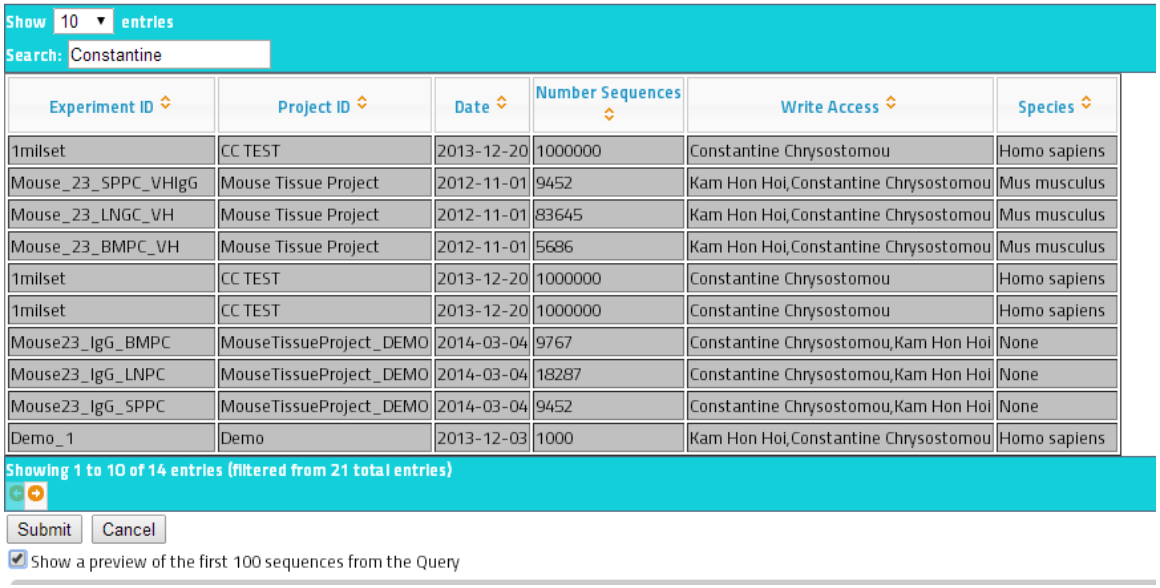
1) Go to the list of APPS by clicking the APPS tab in the top left corner

2) In the search bar, search for the APP called “Download Experiment From Database”

3) Open the APP and select the executive node to use for running the script. In this case we will select the node UTEXAS2.

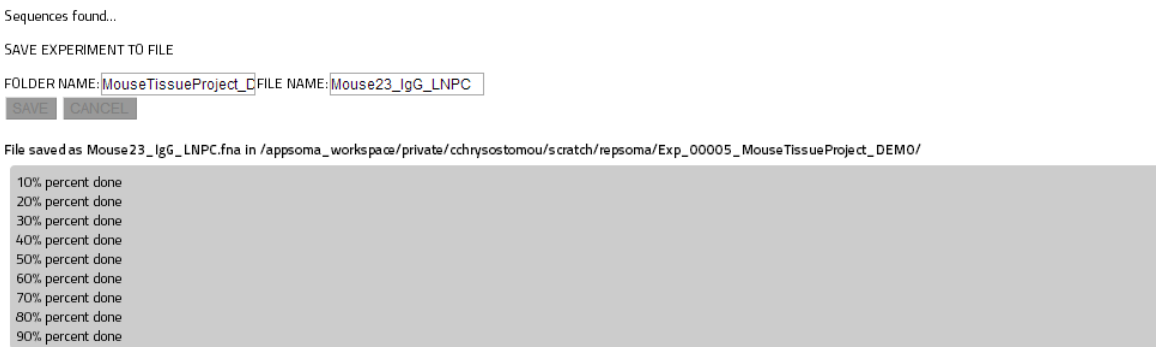
4) The program will list all the experiments present in the database for which you have “Read Access”

Download all sequences associated with a repertoire sequencing experiment.



5) Select the Experiment ID called, Mouse23_IgG_LNPC. Click submit .

6) If the preview table shows that you have selected the correct sequences, then provide the name of the folder and file you want to call this experiment in APPSOMA and press Save.



7) The experiment is automatically added to proper file location within the ImmunoGReP project. Repeat this process for Mouse23_IgG_BMPC and Mouse23_IgG_SPPC.

Steps for annotating sequence data using the ImmunoGReP IgBlast APP on APPSOMA

1) Go to the list of APPS by clicking the APPS tab in the top left corner

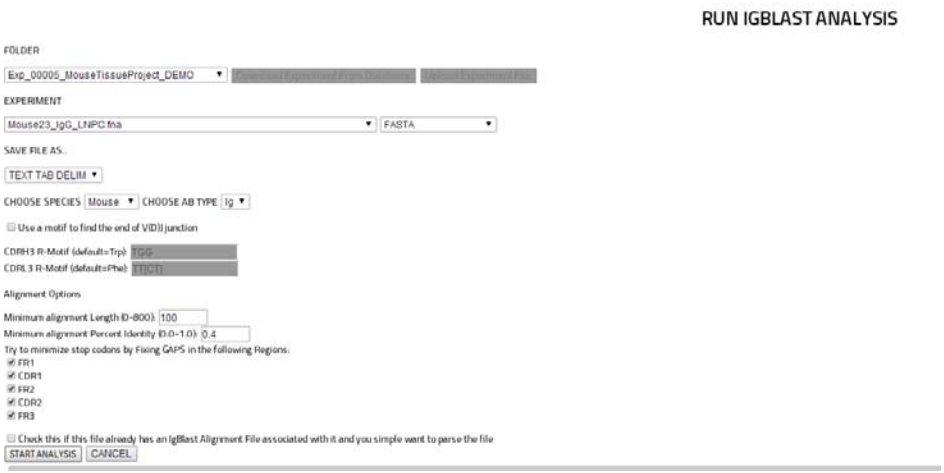
2) In the search bar, search for the APP called “IgBLAST ImmunoAnalysis”

3) Open the APP and select the executive node to use for running the script. In this case we will select the node UTEXAS2.

4) First select the folder location of your file. Next select the corresponding sequence file for your experiment. Supported file types include FASTA file and text-tab-delimited files

5) Fill out the following parameters for this script:

   a. Save File As: choose the file format of the output. Currently supports text tab delimited files and JSON file formats.



b. Species: choose species type. Currently supports mouse and human

c. Use a motif to find end of V(D)J junction: IgBlast is well known to not accurately report the D(J) junction of a CDR3. Therefore, if you would like to use your own motif such as WGXG to represent the end of a CDR3, this script supports that.

d. Minimum alignment length: Select the minimum alignment length required to accurately call an immunoglobulin sequence

e. Minimum percent identity: Select the minimum percent identity to the germline required to accurately call an immunoglobulin sequence.

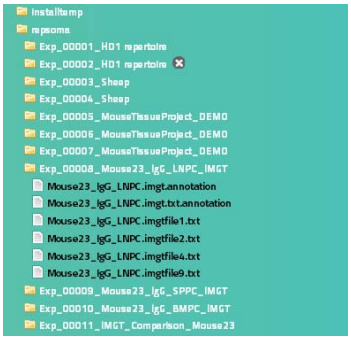
f. Try to minimize stop codons by fixing gaps in regions: Stop codons resulting from gaps detected in the sequences of functional immunoglobulin sequences are often due to experimental error. Therefore, this function tells the program to remove bases which are causing stop codons when aligned to the germline.

g. Check this if the file already has an IgBlast Alignment File: If you have already run IgBlast, but would like to parse the alignment file into the appropriate text or JSON intermediate file for downstream analysis, then click this button.

6) Run the APP. First, the script calls the IgBlast source code and runs an IgBlast alignment file. Then, based on the parameters selected in step [5], it will parse the alignment file and report the result in an easily viewable file that can be opened in excel, if tab-delimited.

7) Repeat steps [1-6] for the next two experiments. Repeat this process for Mouse23_IgG_BMPC and Mouse23_IgG_SPPC.

8) Your analysis files are now saved temporarily in the “scratch” folder of APPSOMA



IMMUNOLOGICAL DATABASE

DATABASE SCHEMA						DESCRIPTION		
Field	**SubDocument 1**	**SubDocument 2**	**SubDocument 3**	**SubDocument 4**	**SubDocument 5**	**Data Structure**	**Description**	**Index**
_Id								
Project_ID						string, not case constant	**This is a string determined by the researcher to describe the current research project (one project will have multiple experiment IDs)**	
Experiment_ID						string, not case constant	**This is a string determined by the researcher to describe his sequencing experiment (one experiment could have multiple sequencing datasets)**	
Date						python datetime format	**Date Experiment was Created**	
Sequencing Platform						string, capitalized	describes how the DNA was sequenced	
Unique_Experiment_ID						python datetime stored as string	**This string stores the date when raw data was uploaded (this should uniquely identify a specific sequencing dataset)**	
Mongo_Experiment_ID						python datetime stored as string	**This string stores the date when raw data is uploaded to MONGO (this should also unique identify a specific sequencing dataset)**	YES
Paired_ID						Mongo _Id datatype	**We will want to link "pairs of documents" that refer to one another, so this string will point to the _id of another document in collection**	YES
Paired						True/False	boolean describes whether this data may have a "paired document"	
Pairing technique						string, capitalized	describes how method used in experiment	
Isotypes_sequenced						list of strings, not case constant	describes what the researcher tried sequencing	
FastQ_Score						string	describes quality of DNA sequence	
Keywords						list of strings, not case constant	keywords for researcher to refer to experiment	
Full_Seq						string, lower case	**raw dna sequence (not analyzed)**	
Chain_Types_sequenced						list of strings, not case constant	Only allow combinations of the follow values: [heavy, light, alpha, and/or beta]	
Seq_Header						string, not case constant	**a unique identifier to refer to the raw DNA sequence**	
Cell_Types_sequenced						list of strings, not case constant	refers to what cells were sequenced	
Write_Access						list of strings referring to first/last name	**names that refers to who has write access to that experiment**	YES
Cell Number						Number	number of cells sequenced	
Species						string	species sequenced	
Read_Access						list of strings referring to first/last name	**refers to people who have read access to that experiment**	YES
Publications						list of strings	publications referring to experiment	
Analysis						Sub Document	**We want to store the results from different ways of analyzing the data so that we can compare each method; ideally we want this database to also be used for making goldset standards for future algorithms**	
	INDEXED_FIELDS					**Sub Document**	**We store redundancy data of all fields that will be indexed in this data structure/sub document; we do this to decrease the number of indexes in our collection**	
		CDR1				List		YES
		LIST ELEMENT 0 =>				Each element of list is subdocument of following		
			AA			string		
			ANAL_TYPE			string		
			RECOMB_TYPE			string		
			NT			string		
		CDR2						YES
		LIST ELEMENT 0 =>				Each element of list is subdocument of following		
			AA			string		
			ANAL_TYPE			string		
			RECOMB_TYPE			string		
			NT			string		
		CDR3						YES
		LIST ELEMENT 0 =>				Each element of list is subdocument of following		
			AA			string		
			ANAL_TYPE			string		
			RECOMB_TYPE			string		
			NT			string		
		V GENES						YES
		LIST ELEMENT 0 =>				Each element of list is subdocument of following		
			ANAL_TYPE			string		
			VGENES			2 element list		
			RECOMB_TYPE			string		
		D GENES						YES

Table D.3: Document schema of Sequences Collection

DATABASE SCHEMA						DESCRIPTION		
Field	SubDocument 1	SubDocument 2	SubDocument 3	SubDocument 4	SubDocument 5	Data Structure	Description	Index
		LIST ELEMENT 0 =>				Each element of list is subdocument of following		
			ANAL_TYPE			string		
			D GENES			2 element list		
			RECOMB_TYPE			string		
		J GENES						YES
		LIST ELEMENT 0 =>				Each element of list is subdocument of following		
			ANAL_TYPE			string		
			J GENES			2 element list		
			RECOMB_TYPE			string		
		PRODUCTIVE				boolean true/false		YES
		LIST ELEMENT 0 =>				Each element of list is subdocument of following		
			ANAL_TYPE			string		
			PROD			string		
			RECOMB_TYPE			string		
		CHAIN_TYPE				string		YES
		LIST ELEMENT 0 =>				Each element of list is subdocument of following		
			ANAL_TYPE			sting		
			CHAIN			string		
	IMGT_ANALYSIS					**SubDocument**	**This is one algorithm we use to anlayze the data**	
		PREDICTED_CHAIN_TYPE				list of strings	only options: [heavy,light,alpha,beta]	
		PREDICTED_ISOTYPE				string, not case constant		
		PRODUCTIVE				string, not case constant		
		VDJ				SubDocument		
			LOCUS_NAME			string, case constant	Only IgH, or TrB	
			VREGION			SubDocument		
				SHM		string, case constant		
				FR1		SubDocument		
					NT	string, lower case	this is a nucleotide dna sequence	
					AA	string, upper case	this is the amino acid sequence of the nucleotide sequence above	
				FR2		SubDocument		
					NT	string, lower case		
					AA	string, upper case		
				FR3		SubDocument		
			CDR1		NT	string, lower case		
			NT		AA	string, upper case		
			CDR2	CDR1		SubDocument		
			NT		NT	string, lower case		
			VGenes		AA	string, upper case		
				CDR2		SubDocument		
					NT	string, lower case		
					AA	string, upper case		
				Vgenes		List of strings	i.e. [IGHV13*01,IGHV31*01]	
			CDR3			SubDocument		
				NT		string, lower case		
				AA		string, upper case		
			DREGION			SubDocument		
				Dgenes		list of strings	i.e. [IGHV13*01,IGHV31*01]	
			JREGION			SubDocument		
				FR4		SubDocument		
					NT	list of strings		
					AA	list of strings		
		VJ				SubDocument		
			LOCUS_NAME			string, case constant	Only IgH, or TrB	
			VREGION			SubDocument		
				SHM		string, case constant		
				FR1		SubDocument		
					NT	string, lower case		
					AA	string, upper case		
				FR2		SubDocument		
					NT	string, lower case		
					AA	string, upper case		
				FR3		SubDocument		
					NT	string, lower case		
					AA	string, upper case		
				CDR1		SubDocument		
					NT	string, lower case		
					AA	string, upper case		
				CDR2		SubDocument		
					NT	string, lower case		
					AA	string, upper case		
				Vgenes		List of strings	i.e. [IGHV13*01,IGHV31*01]	
			CDR3			SubDocument		
				NT		string, lower case		
				AA		string, upper case		
			JREGION			SubDocument		
				FR4		SubDocument		
					NT	list of strings		
					AA	list of strings		
	IGBLAST_ANALYSIS					SubDocument	**This is anoter program we use for analyzing the same data; they yield slightly different results for different fields**	
		PREDICTED_CHAIN_TYPE				list of strings	only options: [heavy,light,alpha,beta]	
		PREDICTED_ISOTYPE				string, not case constant		
		PRODUCTIVE				string, not case constant		
		VDJ				SubDocument		
			LOCUS_NAME			string, case constant	Only IgH, or TrB	
			VREGION			SubDocument		
				SHM		string, case constant		

Table D.3 continued

DATABASE SCHEMA						DESCRIPTION		
Field	SubDocument 1	SubDocument 2	SubDocument 3	SubDocument 4	SubDocument 5	Data Structure	Description	Index
				FR1		SubDocument		
					NT	string, lower case		
					AA	string, upper case		
				FR2		SubDocument		
					NT	string, lower case		
					AA	string, upper case		
				FR3		SubDocument		
			CDR1		NT	string, lower case		
			NT		AA	string, upper case		
			CDR2	CDR1		SubDocument		
			NT		NT	string, lower case		
			VGenes		AA	string, upper case		
				CDR2		SubDocument		
					NT	string, lower case		
					AA	string, upper case		
				Vgenes		List of strings	i.e. [IGHV13*01,IGHV31*01]	
			CDR3			SubDocument		
				NT		string, lower case		
				AA		string, upper case		
			DREGION			SubDocument		
				Dgenes		list of strings	i.e. [IGHV13*01,IGHV31*01]	
			JREGION			SubDocument		
				FR4		SubDocument		
					NT	list of strings		
					AA	list of strings		
		VJ				SubDocument		
			LOCUS_NAME			string, case constant	Only IgH, or TrB	
			VREGION			SubDocument		
				SHM		string, case constant		
				FR1		SubDocument		
					NT	string, lower case		
					AA	string, upper case		
				FR2		SubDocument		
					NT	string, lower case		
					AA	string, upper case		
				FR3		SubDocument		
					NT	string, lower case		
					AA	string, upper case		
				CDR1		SubDocument		
					NT	string, lower case		
					AA	string, upper case		
				CDR2		SubDocument		
					NT	string, lower case		
					AA	string, upper case		
				Vgenes		List of strings	i.e. [IGHV13*01,IGHV31*01]	
			CDR3			SubDocument		
				NT		string, lower case		
				AA		string, upper case		
			JREGION			SubDocument		
				FR4		SubDocument		
					NT	list of strings		
					AA	list of strings		
	INHOUSE_ANALYSIS					SubDocument	**Eventually we want to compare both methods above to our inhouse programs**	
		PREDICTED_CHAIN_TYPE				list of strings	only options: [heavy,light,alpha,beta]	
		PREDICTED_ISOTYPE				string, not case constant		
		PRODUCTIVE				string, not case constant		
		VDJ				SubDocument		
			LOCUS_NAME			string, case constant	Only IgH, or TrB	
			VREGION			SubDocument		
				SHM		string, case constant		
				FR1		SubDocument		
					NT	string, lower case		
					AA	string, upper case		
				FR2		SubDocument		
					NT	string, lower case		
					AA	string, upper case		
				FR3		SubDocument		
			CDR1		NT	string, lower case		
			NT		AA	string, upper case		
			CDR2	CDR1		SubDocument		
			NT		NT	string, lower case		
			VGenes		AA	string, upper case		
				CDR2		SubDocument		
					NT	string, lower case		
					AA	string, upper case		
				Vgenes		List of strings	i.e. [IGHV13*01,IGHV31*01]	
			CDR3			SubDocument		
				NT		string, lower case		
				AA		string, upper case		
			DREGION			SubDocument		
				Dgenes		list of strings	i.e. [IGHV13*01,IGHV31*01]	
			JREGION			SubDocument		
				FR4		SubDocument		
					NT	list of strings		
					AA	list of strings		
		VJ				SubDocument		
			LOCUS_NAME			string, case constant	Only IgH, or TrB	
			VREGION			SubDocument		
				SHM		string, case constant		
				FR1		SubDocument		
					NT	string, lower case		
					AA	string, upper case		
				FR2		SubDocument		
					NT	string, lower case		
					AA	string, upper case		
				FR3		SubDocument		
					NT	string, lower case		
					AA	string, upper case		
				CDR1		SubDocument		
					NT	string, lower case		
					AA	string, upper case		
				CDR2		SubDocument		
					NT	string, lower case		
					AA	string, upper case		
				Vgenes		List of strings	i.e. [IGHV13*01,IGHV31*01]	

Table D.3 continued

DATABASE SCHEMA						DESCRIPTION		
Field	SubDocument 1	SubDocument 2	SubDocument 3	SubDocument 4	SubDocument 5	Data Structure	Description	Index
			CDR3			SubDocument		
				NT		string, lower case		
				AA		string, upper case		
			JREGION			SubDocument		
				FR4		SubDocument		
					NT	list of strings		
					AA	list of strings		

Table D.3 continued

DATABASE SCHEMA	DESCRIPTION		
Field	**Data Structure**	Description	**Desired Indexed**
PROJECT_ID:	string	This is a string determined by the researcher to describe the current research project (one project will have multiple experiment IDs)	
PROJECT_ID_INDEX:	string	same as project id, except we remove all white spaces and lower case the string	yes
EXPERIMENT_ID:	string	This is a string determined by the researcher to describe his sequencing experiment (one experiment could have multiple sequencing datasets)	
EXPERIMENT_ID_INDEX:	string	same as experiment id, except we remove all white spaces and lower case the string	yes
MONGO_EXPERIMENT_ID	python datetime stored as string	This string stores the date when raw data is uploaded to MONGO (this should also unique identify a specific sequencing dataset)	
UNIQUE_EXPERIMENT_ID:	python datetime stored as string	This string stores the date when raw data was uploaded (this should uniquely identify a specific sequencing dataset)	yes
DATE:	python datetime format	Date Experiment was Created	
SEQUENCING_PLATFORM:	list of strings	describes how the DNA was sequenced	yes
PAIRED:	boolean	Describes whether Antibodies were paired in this experiment	
PARING_TECHNIQUE:	string	describes how method used in experiment	
PAIRING_TECHNIQUE_INDEX:	string	Same as pairing_technique except removed all white spaces and lower case the string	yes
ISOTYPE:	list of strings	describes what the researcher tried sequencing	
ISOTYPE_INDEX:	list of strings	Same as isotype except removed all white spaces and lower case the string	yes
CHAIN_TYPES_SEQUENCED:	list of strings	list of the following options only: [heavy, light, alpha, beta]	yes
CELL_TYPES_SEQUENCED:	list of strings	refers to what cells were sequenced	
CELL_TYPE_INDEX:	list of strings	same as "cell_types_sequenced" except removed all white spaces and lower cased the string	yes
WRITE_ACCESS:	list of strings	names that refers to who has write access to that experiment	
WRITE_ACCESS_INDEX:	list of strings	same as "write_access" except removed all white spaces and lower cased the string	yes
SPECIES:	list of strings	species sequenced	yes
READ_ACCESS:	list of strings	refers to people who have read access to that experiment	
READ_ACCESS_INDEX:	list of strings	same as "read_access" except removed all white spaces and lower cased the string	yes
PUBLICATIONS:	list of strings	publications referring to experiment	
PUBLICATIONS_INDEX:	list of strings	same as "publications" except removed all white spaces and lower cased the string	yes

Table D.4: Document schema of experiment collection

DATABASE SCHEMA		DESCRIPTION		
Field	**SubDocument 1**	**Data Structure**	**Description**	Example
_id		String		ObjectId("52212fef1d41c8233d2ede07")
Locus		String	Bcell H/L or T cell A/B chain	IGH
Sublocus		String	V, D, or J region	J-REGION
Method		String	database source	IMGT
Species		String	species name	Homo sapiens
Sequences	[each element of list is subdocument]	List of subdocuments	list of all genes within this subset	
	Germline Family	string	name of germline family	IGHJ1
	stat and end position	string	position along chromosome	723...724
	Sequence	string	actual nucleotide sequence of	gctacgg....
	IMGT Accession number	string	IMGT reference	J00256
	Exon(s), region names, or extracted label(s)	string	V, D, or J region	J-REGION
	Functionality	string	F for funcional, P for pseudogene, ORF for open reading frame	F
	Germline Gene Name	string	germline gene name for sequence	IGHJ1
	Germline Gene Allele Name	string	allelic name of germline gene	IGHJ1*01
	Species	string	species name	Homo sapiens

Table D.5: Document schema of germline collection

Description of APPSOMA scripts used for developing the ImmunoGReP pipeline

All code for scripts are openly available on the APPSOMA webpage. The following summarize the functions we provide.

Filename: cchrysostomou_databaseschema.py

Description: stores all relevant functions for using the database schema. Contains functions which report the default structure of the database in the form of a dictionary.

File type: python code

Functions:

DB_RequiredFields: List of required fields for inserting into sequences database

DB_OptionalFields: List of optional fields for inserting documents into sequences database

DB_AnalysisIMGT: List of fields returned by IMGT analysis

DB_AnalysisIGBLAST: List of fields returned by IgBlast analysis

DB_AnalysisINHOUSE: List of fields returned by InHouse analysis

convertDictionaryIntoMongoSequence: Takes a python dictionary of terms, usually metadata from IRODs, and formats them so that they are properly inserted into the experiments and sequences collection

convert_metadata_from_irods_to_dic: Takes in a file name corresponding to an output from an IRODs query. This file lists all metadata assocated with the query. This function will convert the text file into a python dictionary.

removeNoneVals: Removes "null" values from dictionary

extract_DB_INFO: When getting a fasta file from the database, many values are appended to the sequence header which correspond to its location in the database. This function, unwraps the database specific variables from the sequence header and appends them in another variable.

removeFileExtension: This function removes the file extension from a string file name

Filename: cchrysostomou_readwritefiles.py

Description: Most of our scripts use one of the following formats: text tab delimited files, JSON formatted files, and FASTA files. In addition, with each of our file formats, we output special "@" symbols which are usually specific for developers to know where the files came from and how to handle inserting files back into the database. Therefore, given our unique file formats, we have written a series of functions for reading and writing files for ImmunoGReP platform.

Filetype: python code

Functions:

previewJsonFields: returns a list of all the fields or keys in the json file
previewTxtFields: returns a list of the header row from a text file
checkFileType: returns whether a file is a txt file, fasta file, or a JSON file
TXT_to_JSON: converts a text-tab-delimited file to a JSON file
JSON_to_TXT: converts a JSON file to a text-tab-delimited file
JSON_to_Dictionary: opens a JSON file as a list of dictionaries in python
TXT_to_Dictionary: opens a text file as a list of dictionaries in python
Dictionary_to_TXT: converts a python list of dictionaries to a text file
Dictionary_to_JSON: converts a python list of dictionaries to a JSON file

Filename: andrew_ig_tools.py

Description: This defines a class of functions used to traverse the folder structure of ImmunoGReP

Filetype: python code

Functions:

_projName: global variable defining the name of the project folder (i.e. immunogrep)
expDirParams: global variable defining the general structure of folder names (currently set as: Exp#####_projectname or date)
dir_files: lists all files within a parent folder
get_all_files_recursively: lists all files with all subfolders of a parent folder
class ExperimentDirs:
get_basedir: return the base directory
get_dirs: list all directories in the immunogrep path

make_dir: make a new directory within immunogrep based on the structure of folder names

Filename: cchrysostomou_add_fasta_to_db.py

Description: Creates a GUI table listing all experiments a user has uploaded to IRODs

Filetype: python and HTML

Functions:

GenerateIRODSTable: Generate HTML code that creates a java datatable using metadata downloaded from IRODs

addFastaToDB: 1) Generate the IRODs table, 2) wait for user to select an experiment to download, 3) download the fasta file for that selected experiment, 4) if present, download the IMGT file for that selected experiment, 5) create a file containing both sequence and IMGT information, 6) add file to database

Filename: cchrysostomou_add_seq_irods_to_mongo.py

Description: Contains functions for adding data to the experiments and sequences collections in the database

Filetype: python

Functions:

addSeqToDatabase: Add only a fasta file containing a sequence and sequence header to the database

addNewSeqIMGTToDatabase: If IMGT data is also associated with the fasta file, then add both the sequence information in the fasta file and its corresponding IMGT information as defined by DB_AnalysisIMGT

Filename: cchrysostomou_ linkimgttofasta.py

Description: this is a single function that links sequences within a fasta file to their corresponding IMGT analysis. This does not assume that the row order of sequences in the fasta file matches the order of results/sequences reported by IMGT. Instead, it links the sequence header each sequence in a fasta file with the sequence header stored after an IMGT analysis

Filetype: python

Filename: kamhonhoi_install_icommands.bash

Description: In order to use IRODs and parse through stored files, you need to first install icommands. This script automatically installs icommands for you.

Filetype: bash

Filename: cchrysostomou_igblast_install.bash

Description: This script automatically installs both blast, igblast, and any germline library files required by igblast to your APPSOMA directory

Filetype: bash

Filename: cchrysostomou_run_igblast_command.py

Description: This script is used to run IgBlast from python. All variables and parameters for IgBlast are defined in a python dictionary called variable_parameters. Those values are passed to the IgBlast command.

Filetype: python

Filename: cchrysostomou_parse_igblastfile.py

Description: This script will first invoke run_igblast_command function and run an igblast analysis on a fasta file. Once igblast is complete, this script will open the igblast file and parse it into a easily interpretable tab-delimited, or json output containing relevant immunoglobulin features.

Filetype: python

Functions:

Parse_IgBlast_File: actually parses the igblast file (assumes igblast outputs -7 file format). All relevant features such as VGENES and CDR1 are stored in a dictionary. This list of dictionary values are written to a text file at the end.

DefaultIgBlastDic: This is the default format for an IgBlast analysis stored as a dictionary

Write_Seq_JSON: Writes the output results for a sequence to a JSON file
TABFileHeader: Dictionary that points column names to actual column number in a tab delimited file
DatabaseTranslator: This is an essential function for users who would like to update the database with an IgBlast analysis. This writes a special line to the output format that is detected specifically by later functions for inserting back into the database. This gives direction with respect to which fields should be inserted and where in the database.
WriteTABFileHeader: Writes the header row for a tab-delimited output result
GetAlignment: Takes in the alignment between the germline and sequence of interest. It translates the sequence of interest into its amino acid sequence. If (1) a stop codon is encountered, (2) the user would like to correct stop codons, and (3) a gap is detected in the alignment, then the stop codon will attempt to be fixed by fixing potential insertions and deletions.
Fix_Ins_Del: If this function is called, then it will attempt to repair insertion/deletions. Takes the alignment of the germline to the sequence of interest and if there is a gap between the sequences, this will remove the insertion/deletion error
Check_Frame: Checks the current frame of the antibody nucleotide sequence. If it is out of frame, i.e. the nucleotide sequence starts at position 2 rather than 1, it adds, "N's" to the end to put the sequence back in frame.

Filename: cchrysostomou_coutnuniquevalues.py

Description: this function will take either a text file or JSON file and count the number of times unique values are encountered in predefined field name or column number

Filetype: python

Filename: bennigoetz_experiment_to_sequences.py

Description: this function will download all sequences which correspond to a specific experiment in the Mongo database. The sequences are downloaded and written to a FASTA file.

Filetype: python and HTML
Functions:

MakeExpTable: Creates an interactive table listing all experiments currently in the database

MakeDocResultsTable: Creates an interactive table listing the first 100 sequences within an experiment
SaveFile: Saves the result as a new fasta file in the ImmunoGReP folder structure. This file can be used for downstream analyses.

Filename: kamhonhoi_cdr3.py

Description: Runs the PSSM search for DNA motifs that flank the CDR3

Filetype: python

Filename: kamhonhoi_appsoma_cdr3tools.py

Description: This contains supplemental functions for running the PSSM CDR3 search.

Filetype: python

Functions:

MotifDB2JSON: Download the motif table from the database into a JSON file
MotifTXT2JSON: Convert a text tab delimited file containing a motif into a JSON file
convertMotifTable2Var: converts a motif table into variables used for the PSSM search

Filename: kamhonhoi_taxonomy_tools.py

Description: This file contains a list of functions for querying the NCBI taxonomy ID for selected species within the database

Filtype: python
Functions:

getTaxID: Given a string name for a species, this returns the taxonomy ID reported by NCBI
pushTaxID2DB: update the species collection with a new species taxonomy
convert2Species: this function will convert a taxonomy ID into a species name

Filename: bennigoetz_kams_cdr3circosgencomplete.pl

Description: this script generates a circus plot given an input file of the correct format. The input format is one text tab delimited file with Nxfour columns for each Nxdatasets you would like to compare using a circus plot. The first three columns for each dataset correspond to string you would like to compare such as CDR3, count, and frequency. The fourth column is blank. The first row of the file will containing the name of each dataset separated by tabs.

Filetype: Perl

Filename: sschaetzle_comparethingy.py

Description: This script will compare a certain field such as CDR3 sequence across multiple datasets. First it reports the number of total unique sequences or strings observed and the number of times that sequence is observed in each dataset. Then it calculates the pairwise spearman and pearson correlations for each dataset pair. Finally, it groups each sequence based on which compartments or datasets it was observed in. For example, if comparing 3 datasets, it will group the data into 7 compartments which represent all possible combinations of overlap. At the end, it plots the correlation as a heatmap.

Filetype: Python

Filename: bennigoetz_db_ops.py

Description: This script defines functions relevant to inserting documents into the database

Filetype: Python

Functions:

index_fields: global variables that defines what fields will be indexed in the database

analysis_types: global variable that defines the types of analysis currently in the database
recomb_types: global variable defining either heavy chain ('VDJ') or light chain ('VJ') analyses
convert_text_to_index_field_text: converts the value for that field in the document to a format that is amenable for queries. For example, it removes whitespaces and case-sensitive characters.
Find_indexed_fields_in_multilevel_dict: Generate dictionaries for sequence index subdocument. This is a generator function that recursively finds ANALYSIS_TYPE, NT, AA, RECOMB_TYPE and yields a dictionary whenever it gathers enough information to fill out a dictionary with indexed values.
generate_indexed_fields_from_sequence_dict: Helper function for the indexed_fields generator. The generator spits out the dictionaries for the INDEXED_FIELDS field. This function accumulates the output from the generator and stores it in a dictionary {'INDEXED_FIELDS': subdictionaries} which it returns
return_nested_values_with_dotted_keys: Search through nested dictionary; keep track of nested keys using dot notation (and the custom DotAccessible class). When a non-dictionary value is hit, yield the dotted key-value pair. This is a recursive generator function.
flatten_multilevel_dict_with_dots: Accumulate the one-element dictionaries from the return_nested_values_with_dotted_keys generator function into a single dictionary.
generate_new_subdictionaries: Search through nested dictionary; keep track of nested keys using dot notation (and the custom DotAccessible class). When a non-dictionary value is hit, yield the dotted key-value pair. This is a recursive generator function.
find_new_subdictionaries: Accumulate the one-element dictionaries from the return_nested_values_with_dotted_keys generator function into a single dictionary.
update_experiments: Updates the experiments collection given the data, for either an insert or an update.
insert_record: Create a new record. This function checks that all required fields for a sequence document are present. It then inserts the "data" dictionary into the sequences collection. Then it creates relevant INDEXED fields.
update_record: Given a $oid, this function updates the sequence and experiments collections with the given data.
save_record: Insert and update wrapper. Should return an $oid or a list of $oids of records modified/created. If insert=True, it calls insert_record to insert a new record. When update_record is called, any data already in the

database is replaced. The append argument is only relevant for update operations on arrays. Rather than replacing an array, it appends to the array.

References

1. Murphy, K., Travers, Paul & Walport, Mark. *Janeway's Immunobiology*. (Taylor & Francis, 2007).
2. Barquet, N. & Domingo, P. Smallpox: the triumph over the most terrible of the ministers of death. *Ann. Intern. Med.* **127,** 635–642 (1997).
3. Nelson, A. L., Dhimolea, E. & Reichert, J. M. Development trends for human monoclonal antibody therapeutics. *Nat. Rev. Drug Discov.* **9,** 767–774 (2010).
4. Norman, P. Monoclonal Antibodies in the Pipeline: A Segment of Major Growth. *Camb. Heal. Inst.* (2011).
5. Wang, Wei, Singh, S., Zeng, D. L., King, Kevin & Nema, Sandeep. Antibody structure, instability, and formulation. *J. Pharm. Sci.* **96,** 1–26 (2007).
6. Hunter, R. L. Overview of vaccine adjuvants: present and future. *Vaccine* **20, Supplement 3,** S7–S12 (2002).
7. Guy, B. The perfect mix: recent progress in adjuvant research. *Nat. Rev. Microbiol.* **5,** 505–517 (2007).
8. Georgiou, G. *et al.* The promise and challenge of high-throughput sequencing of the antibody repertoire. *Nat. Biotechnol.* **advance online publication,** (2014).
9. Reddy, S. T. *et al.* Monoclonal antibodies isolated without screening by analyzing the variable-gene repertoire of plasma cells. *Nat. Biotechnol.* **28,** 965–969 (2010).
10. Wine, Y. *et al.* Molecular deconvolution of the monoclonal antibodies that comprise the polyclonal serum response. *Proc. Natl. Acad. Sci.* 201213737 (2013). doi:10.1073/pnas.1213737110
11. Weinstein, J. A., Jiang, N., White, R. A., Fisher, D. S. & Quake, S. R. High-Throughput Sequencing of the Zebrafish Antibody Repertoire. *Science* **324,** 807–810 (2009).
12. Lavinder, J. J. *et al.* Identification and characterization of the constituent human serum antibodies elicited by vaccination. *Proc. Natl. Acad. Sci.* 201317793 (2014). doi:10.1073/pnas.1317793111
13. Apostoaei, A.J. & Trabalka, J.R. Review, Synthesis, and Application of Information on the Human Lymphatic System to Radiation Dosimetry for Chronic Lymphocytic Leukemia.
14. Savina, A. & Amigorena, S. Phagocytosis and antigen presentation in dendritic cells. *Immunol. Rev.* **219,** 143–156 (2007).
15. Poljak, R. J. *et al.* Three-dimensional structure of the Fab' fragment of a human immunoglobulin at 2,8-A resolution. *Proc. Natl. Acad. Sci. U. S. A.* **70,** 3305–3310 (1973).
16. Ramsland, P. A. & Farrugia, W. Crystal structures of human antibodies: a detailed and unfinished tapestry of immunoglobulin gene products. *J. Mol. Recognit.* **15,** 248–259 (2002).
17. Chaudhuri, J. & Alt, F. W. Class-switch recombination: interplay of transcription, DNA deamination and DNA repair. *Nat. Rev. Immunol.* **4,** 541–552 (2004).

18. Jefferis, R. & Kumararatne, D. S. Selective IgG subclass deficiency: quantification and clinical relevance. *Clin. Exp. Immunol.* **81,** 357–367 (1990).
19. Cerutti, A. The regulation of IgA class switching. *Nat. Rev. Immunol.* **8,** 421–434 (2008).
20. Gould, H. J. & Sutton, B. J. IgE in allergy and asthma today. *Nat. Rev. Immunol.* **8,** 205–217 (2008).
21. Tonegawa, S. Somatic generation of antibody diversity. *Nature* **302,** 575–581 (1983).
22. Berek, C., Berger, A. & Apel, M. Maturation of the immune response in germinal centers. *Cell* **67,** 1121–1129 (1991).
23. Kim, S., Davis, M., Sinn, E., Patten, P. & Hood, L. Antibody diversity: somatic hypermutation of rearranged VH genes. *Cell* **27,** 573–581 (1981).
24. Walter, M. A., Surti, U., Hofker, M. H. & Cox, D. W. The physical organization of the human immunoglobulin heavy chain gene complex. *EMBO J.* **9,** 3303–3313 (1990).
25. Frippiat, J. P. *et al.* Organization of the human immunoglobulin lambda light-chain locus on chromosome 22q11.2. *Hum. Mol. Genet.* **4,** 983–991 (1995).
26. Kawasaki, K. *et al.* Evolutionary dynamics of the human immunoglobulin kappa locus and the germline repertoire of the Vkappa genes. *Eur. J. Immunol.* **31,** 1017–1028 (2001).
27. Hozumi, N. & Tonegawa, S. Evidence for somatic rearrangement of immunoglobulin genes coding for variable and constant regions. *Proc. Natl. Acad. Sci. U. S. A.* **73,** 3628–3632 (1976).
28. Lefranc, M.-P. IMGT, the international ImMunoGeneTics information system http:www.imgt.org.
29. Oettinger, M. A., Schatz, D. G., Gorka, C. & Baltimore, D. RAG-1 and RAG-2, adjacent genes that synergistically activate V(D)J recombination. *Science* **248,** 1517–1523 (1990).
30. Kapitonov, V. V. & Jurka, J. RAG1 core and V(D)J recombination signal sequences were derived from Transib transposons. *PLoS Biol.* **3,** e181 (2005).
31. Osmond, D. G., Rolink, A. & Melchers, F. Murine B lymphopoiesis: towards a unified model. *Immunol. Today* **19,** 65–68 (1998).
32. Meier, J. T. & Lewis, S. M. P nucleotides in V(D)J recombination: a fine-structure analysis. *Mol. Cell. Biol.* **13,** 1078–1092 (1993).
33. MacLennan, I. C. M. *et al.* Extrafollicular antibody responses. *Immunol. Rev.* **194,** 8–18 (2003).
34. Zotos, D. & Tarlinton, D. M. Determining germinal centre B cell fate. *Trends Immunol.* **33,** 281–288 (2012).
35. D A Fulcher, A. B. B cell life span: a review. *Immunol. Cell Biol.* **75,** 446–55 (1997).
36. Gatto, D. & Brink, R. The germinal center reaction. *J. Allergy Clin. Immunol.* **126,** 898–907 (2010).

37. Victora, G. D. & Nussenzweig, M. C. Germinal centers. *Annu. Rev. Immunol.* **30,** 429–457 (2012).
38. Nagaoka, H., Muramatsu, M., Yamamura, N., Kinoshita, K. & Honjo, T. Activation-induced Deaminase (AID)-directed Hypermutation in the Immunoglobulin S? Region. *J. Exp. Med.* **195,** 529–534 (2002).
39. Teng, G. & Papavasiliou, F. N. Immunoglobulin Somatic Hypermutation. *Annu. Rev. Genet.* **41,** 107–120 (2007).
40. Li, Z., Woo, C. J., Iglesias-Ussel, M. D., Ronai, D. & Scharff, M. D. The generation of antibody diversity through somatic hypermutation and class switch recombination. *Genes Dev.* **18,** 1–11 (2004).
41. Goding, J. W. Antibody production by hybridomas. *J. Immunol. Methods* **39,** 285–308 (1980).
42. Lanzavecchia, A. & Sallusto, F. Human B cell memory. *Curr. Opin. Immunol.* **21,** 298–304 (2009).
43. Li, J. *et al.* Human antibodies for immunotherapy development generated via a human B cell hybridoma technology. *Proc. Natl. Acad. Sci. U. S. A.* **103,** 3557–3562 (2006).
44. Tomita, M. & Tsumoto, K. Hybridoma technologies for antibody production. *Immunotherapy* **3,** 371–380 (2011).
45. Tiller, T. Single B cell antibody technologies. *New Biotechnol.* **28,** 453–457 (2011).
46. Wardemann, H. *et al.* Predominant autoantibody production by early human B cell precursors. *Science* **301,** 1374–1377 (2003).
47. Isnardi, I. *et al.* IRAK-4- and MyD88-Dependent Pathways Are Essential for the Removal of Developing Autoreactive B Cells in Humans. *Immunity* **29,** 746–757 (2008).
48. Reddy, S. T. & Georgiou, G. Systems analysis of adaptive immunity by utilization of high-throughput technologies. *Curr. Opin. Biotechnol.* **22,** 584–589 (2011).
49. Wrammert, J. *et al.* Rapid cloning of high-affinity human monoclonal antibodies against influenza virus. *Nature* **453,** 667–671 (2008).
50. Meijer, P.-J. *et al.* Isolation of human antibody repertoires with preservation of the natural heavy and light chain pairing. *J. Mol. Biol.* **358,** 764–772 (2006).
51. Wiberg, F. C. *et al.* Production of target-specific recombinant human polyclonal antibodies in mammalian cells. *Biotechnol. Bioeng.* **94,** 396–405 (2006).
52. Smith, K. *et al.* Rapid generation of fully human monoclonal antibodies specific to a vaccinating antigen. *Nat. Protoc.* **4,** 372–384 (2009).
53. Scheid, J. F. *et al.* Broad diversity of neutralizing antibodies isolated from memory B cells in HIV-infected individuals. *Nature* **458,** 636–640 (2009).
54. Wu, X. *et al.* Rational Design of Envelope Identifies Broadly Neutralizing Human Monoclonal Antibodies to HIV-1. *Science* **329,** 856–861 (2010).

55. Klein, F. *et al.* Broad neutralization by a combination of antibodies recognizing the CD4 binding site and a new conformational epitope on the HIV-1 envelope protein. *J. Exp. Med.* **209,** 1469–1479 (2012).
56. Corti, D. *et al.* A Neutralizing Antibody Selected from Plasma Cells That Binds to Group 1 and Group 2 Influenza A Hemagglutinins. *Science* **333,** 850–856 (2011).
57. Simon Tickle, R. A. High-Throughput Screening for High Affinity Antibodies. *J. Assoc. Lab. Autom.* **14,** 303–307 (2009).
58. Sanger, F., Nicklen, S. & Coulson, A. R. DNA sequencing with chain-terminating inhibitors. *Proc. Natl. Acad. Sci. U. S. A.* **74,** 5463–5467 (1977).
59. Soon, W. W., Hariharan, M. & Snyder, M. P. High-throughput sequencing for biology and medicine. *Mol. Syst. Biol.* **9,** 640 (2013).
60. Briney, B. S., Willis, J. R., McKinney, B. A. & Crowe, J. E., Jr. High-throughput antibody sequencing reveals genetic evidence of global regulation of the naïve and memory repertoires that extends across individuals. *Genes Immun.* **13,** 469–473 (2012).
61. Metzker, M. L. Sequencing technologies - the next generation. *Nat. Rev. Genet.* **11,** 31–46 (2010).
62. Mardis, E. R. A decade/'s perspective on DNA sequencing technology. *Nature* **470,** 198–203 (2011).
63. Larimore, K., McCormick, M. W., Robins, H. S. & Greenberg, P. D. Shaping of human germline IgH repertoires revealed by deep sequencing. *J. Immunol. Baltim. Md 1950* **189,** 3221–3230 (2012).
64. Faham, M. *et al.* Deep-sequencing approach for minimal residual disease detection in acute lymphoblastic leukemia. *Blood* **120,** 5173–5180 (2012).
65. Boyd, S. D. *et al.* Individual variation in the germline Ig gene repertoire inferred from variable region gene rearrangements. *J. Immunol. Baltim. Md 1950* **184,** 6986–6992 (2010).
66. Lu, J. *et al.* IgG variable region and VH CDR3 diversity in unimmunized mice analyzed by massively parallel sequencing. *Mol. Immunol.* **57,** 274–283 (2014).
67. Yousfi Monod, M., Giudicelli, V., Chaume, D. & Lefranc, M.-P. IMGT/JunctionAnalysis: the first tool for the analysis of the immunoglobulin and T cell receptor complex V-J and V-D-J JUNCTIONs. *Bioinforma. Oxf. Engl.* **20 Suppl 1,** i379–385 (2004).
68. Ye, J., Ma, N., Madden, T. L. & Ostell, J. M. IgBLAST: an immunoglobulin variable domain sequence analysis tool. *Nucleic Acids Res.* gkt382 (2013). doi:10.1093/nar/gkt382
69. Thompson, J. D., Higgins, D. G. & Gibson, T. J. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.* **22,** 4673–4680 (1994).
70. Edgar, R. C. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.* **32,** 1792–1797 (2004).

71. Ippolito, G. C. *et al.* Antibody Repertoires in Humanized NOD-scid-IL2Rγnull Mice and Human B Cells Reveals Human-Like Diversification and Tolerance Checkpoints in the Mouse. *PLoS ONE* **7,** e35497 (2012).
72. Glanville, J. *et al.* Precise determination of the diversity of a combinatorial antibody library gives insight into the human immunoglobulin repertoire. *Proc. Natl. Acad. Sci.* **106,** 20216–20221 (2009).
73. Larsen, P. A. & Smith, T. P. L. Application of circular consensus sequencing and network analysis to characterize the bovine IgG repertoire. *BMC Immunol.* **13,** 52 (2012).
74. Arnaout, R. *et al.* High-Resolution Description of Antibody Heavy-Chain Repertoires in Humans. *PLoS ONE* **6,** e22365 (2011).
75. Grönwall, C., Pond, S. L. K., Young, J. A. & Silverman, G. J. In Vivo VL-Targeted Microbial Superantigen Induced Global Shifts in the B Cell Repertoire. *J. Immunol.* **189,** 850–859 (2012).
76. Schoettler, N., Ni, D. & Weigert, M. B cell receptor light chain repertoires show signs of selection with differences between groups of healthy individuals and SLE patients. *Mol. Immunol.* **51,** 273–282 (2012).
77. Jiang, N. *et al.* Lineage Structure of the Human Antibody Repertoire in Response to Influenza Vaccination. *Sci. Transl. Med.* **5,** 171ra19–171ra19 (2013).
78. Racanelli, V. *et al.* Antibody Vh Repertoire Differences between Resolving and Chronically Evolving Hepatitis C Virus Infections. *PLoS ONE* **6,** e25606 (2011).
79. Maecker, H. T. *et al.* New tools for classification and monitoring of autoimmune diseases. *Nat. Rev. Rheumatol.* **8,** 317–328 (2012).
80. Parameswaran, P. *et al.* Convergent Antibody Signatures in Human Dengue. *Cell Host Microbe* **13,** 691–700 (2013).
81. Leamon, J. H. *et al.* A massively parallel PicoTiterPlate based platform for discrete picoliter-scale polymerase chain reactions. *Electrophoresis* **24,** 3769–3777 (2003).
82. Kircher, M. & Kelso, J. High-throughput DNA sequencing--concepts and limitations. *BioEssays News Rev. Mol. Cell. Dev. Biol.* **32,** 524–536 (2010).
83. Carlson, C. S. *et al.* Using synthetic templates to design an unbiased multiplex PCR assay. *Nat. Commun.* **4,** (2013).
84. Boyd, S. D. *et al.* Measurement and Clinical Monitoring of Human Lymphocyte Clonality by Massively Parallel V-D-J Pyrosequencing. *Sci. Transl. Med.* **1,** 12ra23–12ra23 (2009).
85. Robins, H. *et al.* Ultra-sensitive detection of rare T cell clones. *J. Immunol. Methods* **375,** 14–19 (2012).
86. Cancro, M. P., Gerhard, W. & Klinman, N. R. The diversity of the influenza-specific primary B-cell repertoire in BALB/c mice. *J. Exp. Med.* **147,** 776–787 (1978).
87. Boyd, S. D. *et al.* Individual Variation in the Germline Ig Gene Repertoire Inferred from Variable Region Gene Rearrangements. *J Immunol* **184,** 6986–6992 (2010).

88. Tonegawa, S. Somatic generation of antibody diversity. *Nature* **302,** 575–581 (1983).
89. Manz, R. A. *et al.* Humoral immunity and long-lived plasma cells. *Curr. Opin. Immunol.* **14,** 517–521 (2002).
90. Yoshida, T. *et al.* Memory B and memory plasma cells. *Immunol. Rev.* **237,** 117–139 (2010).
91. Tonegawa, S., Steinberg, C., Dube, S. & Bernardini, A. Evidence for Somatic Generation of Antibody Diversity. *Proc. Natl. Acad. Sci. U. S. A.* **71,** 4027–4031 (1974).
92. Weigert, M., Gatmaitan, L., Loh, E., Schilling, J. & Hood, L. Rearrangement of genetic information may produce immunoglobulin diversity. *Nature* **276,** 785–790 (1978).
93. Kim, S., Davis, M., Sinn, E., Patten, P. & Hood, L. Antibody diversity: Somatic hypermutation of rearranged VH genes. *Cell* **27,** 573–581 (1981).
94. Berek, C., Berger, A. & Apel, M. Maturation of the immune response in germinal centers. *Cell* **67,** 1121–1129 (1991).
95. Jacob, J., Kelsoe, G., Rajewsky, K. & Weiss, U. Intraclonal generation of antibody mutants in germinal centres. *Nature* **354,** 389–392 (1991).
96. Sanz, I. Multiple mechanisms participate in the generation of diversity of human H chain CDR3 regions. *J. Immunol.* **147,** 1720 –1729 (1991).
97. Victora, G. D. & Nussenzweig, M. C. Germinal centers. *Annu. Rev. Immunol.* **30,** 429–457 (2012).
98. Khurana, S. *et al.* MF59 adjuvant enhances diversity and affinity of antibody-mediated immune response to pandemic influenza vaccines. *Sci. Transl. Med.* **3,** 85ra48 (2011).
99. Wiley, S. R. *et al.* Targeting TLRs expands the antibody repertoire in response to a malaria vaccine. *Sci. Transl. Med.* **3,** 93ra69 (2011).
100. Howard, M. C. Antigen-induced B lymphocyte differentiation. *CRC Crit. Rev. Immunol.* **3,** 181–208 (1982).
101. Reddy, S. T. *et al.* Monoclonal antibodies isolated without screening by analyzing the variable-gene repertoire of plasma cells. *Nat Biotech* **28,** 965–969 (2010).
102. Ravn, U. *et al.* By-passing in vitro screening--next generation sequencing technologies applied to antibody display and in silico candidate selection. *Nucleic Acids Res.* **38,** e193 (2010).
103. Cheung, W. C. *et al.* A proteomics approach for the identification and cloning of monoclonal antibodies from serum. *Nat. Biotechnol.* **30,** 447–452 (2012).
104. Wine, Y. *et al.* Molecular deconvolution of the monoclonal antibodies that comprise the polyclonal serum response. *Proc. Natl. Acad. Sci.* (2013). doi:10.1073/pnas.1213737110
105. Krebber, A. *et al.* Reliable cloning of functional antibody variable domains from hybridomas and spleen cell repertoires employing a reengineered phage display system. *J. Immunol. Methods* **201,** 35–55 (1997).

106. Jost, L. Entropy and diversity. *Oikos* **113,** 363–375 (2006).
107. Soon, W. W., Hariharan, M. & Snyder, M. P. High-throughput sequencing for biology and medicine. *Mol. Syst. Biol.* **9,** (2013).
108. Schroeder, H. W., Jr. Similarity and divergence in the development and expression of the mouse and human antibody repertoires. *Dev. Comp. Immunol.* **30,** 119–135 (2006).
109. Dörner, T. & Lipsky, P. E. Immunoglobulin variable-region gene usage in systemic autoimmune diseases. *Arthritis Rheum.* **44,** 2715–2727 (2001).
110. Kosmas, C. *et al.* Molecular analysis of immunoglobulin genes in multiple myeloma. *Leuk. Lymphoma* **33,** 253–265 (1999).
111. Miklos, J. A., Swerdlow, S. H. & Bahler, D. W. Salivary gland mucosa-associated lymphoid tissue lymphoma immunoglobulin V(H) genes show frequent use of V1-69 with distinctive CDR3 features. *Blood* **95,** 3878–3884 (2000).
112. Yoshida, M. *et al.* Immunoglobulin VH genes in thymic MALT lymphoma are biased toward a restricted repertoire and are frequently unmutated. *J. Pathol.* **208,** 415–422 (2006).
113. Gaëta, B. A. *et al.* iHMMune-align: hidden Markov model-based alignment and identification of germline genes in rearranged immunoglobulin gene sequences. *Bioinforma. Oxf. Engl.* **23,** 1580–1587 (2007).
114. Munshaw, S. & Kepler, T. B. SoDA2: a Hidden Markov Model approach for identification of immunoglobulin rearrangements. *Bioinforma. Oxf. Engl.* **26,** 867–872 (2010).
115. Souto-Carneiro, M. M., Longo, N. S., Russ, D. E., Sun, H. & Lipsky, P. E. Characterization of the Human Ig Heavy Chain Antigen Binding Complementarity Determining Region 3 Using a Newly Developed Software Algorithm, JOINSOLVER. *J. Immunol.* **172,** 6790–6802 (2004).
116. Mardis, E. R. A decade/'s perspective on DNA sequencing technology. *Nature* **470,** 198–203 (2011).
117. Lakhani, K. R. *et al.* Prize-based contests can provide solutions to computational biology problems. *Nat. Biotechnol.* **31,** 108–111 (2013).
118. Lipman, D. J. & Pearson, W. R. Rapid and sensitive protein similarity searches. *Science* **227,** 1435–1441 (1985).
119. Smith, T. F. & Waterman, M. S. Identification of common molecular subsequences. *J. Mol. Biol.* **147,** 195–197 (1981).
120. Frigo, M. & Johnson, S. G. The Design and Implementation of FFTW3. *Proc. IEEE* **93,** 216–231 (2005).
121. Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *J. Mol. Biol.* **215,** 403–410 (1990).
122. Katoh, K., Misawa, K., Kuma, K. & Miyata, T. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Res.* **30,** 3059–3066 (2002).

123. Machado, J. A. T., Costa, A. C. & Quelhas, M. D. Wavelet analysis of human DNA. *Genomics* **98,** 155–163 (2011).
124. Yin, C. & Wang, J. A Novel Method for Comparative Analysis of DNA Sequences by Ramanujan-Fourier Transform. *ArXiv14031523 Cs* (2014). at <http://arxiv.org/abs/1403.1523>
125. Katznelson, Y. *An introduction to harmonic analysis*. (Cambridge University Press, 2004).
126. Cooley, J. W. & Tukey, J. W. An algorithm for the machine calculation of complex Fourier series. *Math. Comput.* **19,** 297–301 (1965).
127. Murtagh, F. Complexities of hierarchic clustering algorithms: State of the art.
128. Briney, B. S., Willis, J. R. & Crowe, J. E. Location and length distribution of somatic hypermutation-associated DNA insertions and deletions reveals regions of antibody structural plasticity. *Genes Immun.* **13,** 523–529 (2012).
129. Wilson, P. C. *et al.* Somatic Hypermutation Introduces Insertions and Deletions into Immunoglobulin V Genes. *J. Exp. Med.* **187,** 59–70 (1998).
130. Giudicelli, V., Chaume, D. & Lefranc, M.-P. IMGT/GENE-DB: a comprehensive database for human and mouse immunoglobulin and T cell receptor genes. *Nucleic Acids Res.* **33,** D256–261 (2005).
131. Hassanieh, H., Indyk, P., Katabi, D. & Price, E. Simple and Practical Algorithm for Sparse Fourier Transform. in *Proc. Twenty-Third Annu. ACM-SIAM Symp. Discrete Algorithms* 1183–1194 (SIAM, 2012). at <http://dl.acm.org/citation.cfm?id=2095116.2095209>
132. Hassanieh, H., Adib, F., Katabi, D. & Indyk, P. Faster GPS via the Sparse Fourier Transform. in *Proc. 18th Annu. Int. Conf. Mob. Comput. Netw.* 353–364 (ACM, 2012). doi:10.1145/2348543.2348587
133. Kidd, B. A., Peters, L. A., Schadt, E. E. & Dudley, J. T. Unifying immunology with informatics and multiscale biology. *Nat. Immunol.* **15,** 118–127 (2014).
134. Benichou, J., Ben-Hamo, R., Louzoun, Y. & Efroni, S. Rep-Seq: uncovering the immunological repertoire through next-generation sequencing. *Immunology* **135,** 183–191 (2012).
135. Baum, P. D., Venturi, V. & Price, D. A. Wrestling with the repertoire: the promise and perils of next generation sequencing for antigen receptors. *Eur. J. Immunol.* **42,** 2834–2839 (2012).
136. Mehr, R., Sternberg-Simon, M., Michaeli, M. & Pickman, Y. Models and methods for analysis of lymphocyte repertoire generation, development, selection and evolution. *Immunol. Lett.* **148,** 11–22 (2012).
137. Zhang, H., Meltzer, P. & Davis, S. RCircos: an R package for Circos 2D track plots. *BMC Bioinformatics* **14,** 244 (2013).
138. Lavinder, J. J. *et al.* Identification and characterization of the constituent human serum antibodies elicited by vaccination. *Proc. Natl. Acad. Sci.* 201317793 (2014). doi:10.1073/pnas.1317793111

139. Goecks, J., Nekrutenko, A. & Taylor, J. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol.* **11,** R86 (2010).
140. Pepke, S., Wold, B. & Mortazavi, A. Computation for ChIP-seq and RNA-seq studies. *Nat. Methods* **6,** S22–S32 (2009).
141. Mesirov, J. P. Accessible Reproducible Research. *Science* **327,** 415–416 (2010).
142. Gentleman, R. C. *et al.* Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol.* **5,** R80 (2004).
143. Brusic, V., Gottardo, R., Kleinstein, S. H., Davis, M. M. & HIPC steering Committee. Computational resources for high-dimensional immune analysis from the Human Immunology Project Consortium. *Nat. Biotechnol.* **32,** 146–148 (2014).
144. Zack Simpson & Ken Demarest. *APPSOMA*. (2012). at <https://appsoma.com/>
145. Rice, P., Longden, I. & Bleasby, A. EMBOSS: the European Molecular Biology Open Software Suite. *Trends Genet. TIG* **16,** 276–277 (2000).
146. Munshaw, S. & Kepler, T. B. SoDA2: a Hidden Markov Model approach for identification of immunoglobulin rearrangements. *Bioinforma. Oxf. Engl.* **26,** 867–872 (2010).
147. Michaeli, M., Noga, H., Tabibian-Keissar, H., Barshack, I. & Mehr, R. Automated cleaning and pre-processing of immunoglobulin gene sequences from high-throughput sequencing. *Front. Immunol.* **3,** 386 (2012).
148. Chen, Z., Collins, A. M., Wang, Y. & Gaëta, B. A. Clustering-based identification of clonally-related immunoglobulin gene sequence sets. *Immunome Res.* **6,** S4 (2010).
149. Mehr, R., Edelman, H., Sehgal, D. & Mage, R. Analysis of mutational lineage trees from sites of primary and secondary Ig gene diversification in rabbits and chickens. *J. Immunol. Baltim. Md 1950* **172,** 4790–4796 (2004).
150. Krzywinski, M. I. *et al.* Circos: An information aesthetic for comparative genomics. *Genome Res.* (2009). doi:10.1101/gr.092759.109
151. Magoč, T. & Salzberg, S. L. FLASH: fast length adjustment of short reads to improve genome assemblies. *Bioinforma. Oxf. Engl.* **27,** 2957–2963 (2011).
152. Introducing JSON. at <http://json.org/>
153. IRODS:Data Grids, Digital Libraries, Persistent Archives, and Real-time Data Systems. at <https://wiki.irods.org/index.php/IRODS:Data_Grids,_Digital_Libraries,_Persistent_Archives,_and_Real-time_Data_Systems>
154. Lou, D. I. *et al.* High-throughput DNA sequencing errors are reduced by orders of magnitude using circle sequencing. *Proc. Natl. Acad. Sci. U. S. A.* **110,** 19872–19877 (2013).
155. Masip, L., Veeravalli, K. & Georgiou, G. The many faces of glutathione in bacteria. *Antioxid. Redox Signal.* **8,** 753–762 (2006).

156. Vlamis-Gardikas, A. The multiple functions of the thiol-based electron flow pathways of Escherichia coli: Eternal concepts revisited. *Biochim. Biophys. Acta* **1780,** 1170–1200 (2008).
157. Gallogly, M. M. & Mieyal, J. J. Mechanisms of reversible protein glutathionylation in redox signaling and oxidative stress. *Curr. Opin. Pharmacol.* **7,** 381–391 (2007).
158. Fahey, R. C. Novel thiols of prokaryotes. *Annu. Rev. Microbiol.* **55,** 333–356 (2001).
159. Jacquot, J.-P. *et al.* Thioredoxins and related proteins in photosynthetic organisms: molecular basis for thiol dependent regulation. *Biochem. Pharmacol.* **64,** 1065–1069 (2002).
160. Messens, J. & Silver, S. Arsenate Reduction: Thiol Cascade Chemistry with Convergent Evolution. *J. Mol. Biol.* **362,** 1–17 (2006).
161. Mukhopadhyay, R., Rosen, B. P., Phung, L. T. & Silver, S. Microbial arsenic: from geocycles to genes and enzymes. *FEMS Microbiol. Rev.* **26,** 311–325 (2002).
162. Maciaszczyk-Dziubinska, E., Wawrzycka, D. & Wysocki, R. Arsenic and Antimony Transporters in Eukaryotes. *Int. J. Mol. Sci.* **13,** 3527–3548 (2012).
163. Kruger, M. C., Bertin, P. N., Heipieper, H. J. & Arsène-Ploetze, F. Bacterial metabolism of environmental arsenic—mechanisms and biotechnological applications. *Appl. Microbiol. Biotechnol.* **97,** 3827–3841 (2013).
164. Rosen, B. P. Biochemistry of arsenic detoxification. *FEBS Lett.* **529,** 86–92 (2002).
165. Ji, G. & Silver, S. Reduction of arsenate to arsenite by the ArsC protein of the arsenic resistance operon of Staphylococcus aureus plasmid pI258. *Proc. Natl. Acad. Sci. U. S. A.* **89,** 9474–9478 (1992).
166. Mukhopadhyay, R., Zhou, Y. & Rosen, B. P. Directed Evolution of a Yeast Arsenate Reductase into a Protein-tyrosine Phosphatase. *J. Biol. Chem.* **278,** 24476–24480 (2003).
167. Bennett, M. S., Guan, Z., Laurberg, M. & Su, X. D. Bacillus subtilis arsenate reductase is structurally and functionally similar to low molecular weight protein tyrosine phosphatases. *Proc. Natl. Acad. Sci. U. S. A.* **98,** 13577–13582 (2001).
168. Liu, J. & Rosen, B. P. Ligand Interactions of the ArsC Arsenate Reductase. *J. Biol. Chem.* **272,** 21084–21089 (1997).
169. Shi, J., Vlamis-Gardikas, A., Åslund, F., Holmgren, A. & Rosen, B. P. Reactivity of Glutaredoxins 1, 2, and 3 fromEscherichia coli Shows That Glutaredoxin 2 Is the Primary Hydrogen Donor to ArsC-catalyzed Arsenate Reduction. *J. Biol. Chem.* **274,** 36039–36042 (1999).
170. DeMel, S., Shi, J., Martin, P., Rosen, B. P. & Edwards, B. F. P. Arginine 60 in the ArsC arsenate reductase of E. coli plasmid R773 determines the chemical nature of the bound As(III) product. *Protein Sci. Publ. Protein Soc.* **13,** 2330–2340 (2004).
171. Veeravalli, K., Boyd, D., Iverson, B. L., Beckwith, J. & Georgiou, G. Laboratory evolution of glutathione biosynthesis reveals natural compensatory pathways. *Nat. Chem. Biol.* **7,** 101–105 (2011).

172. Kitagawa, M. *et al.* Complete set of ORF clones of Escherichia coli ASKA library (A Complete Set of E. coli K-12 ORF Archive): Unique Resources for Biological Research. *DNA Res.* **12,** 291–299 (2006).
173. Patrick, W. M., Quandt, E. M., Swartzlander, D. B. & Matsumura, I. Multicopy Suppression Underpins Metabolic Evolvability. *Mol. Biol. Evol.* **24,** 2716–2722 (2007).
174. Patrick, W. M. & Matsumura, I. A study in molecular contingency: glutamine phosphoribosylpyrophosphate amidotransferase is a promiscuous and evolvable phosphoribosylanthranilate isomerase. *J. Mol. Biol.* **377,** 323–336 (2008).
175. Desai, K. K. & Miller, B. G. Recruitment of genes and enzymes conferring resistance to the nonnatural toxin bromoacetate. *Proc. Natl. Acad. Sci.* **107,** 17968–17973 (2010).
176. Nishida, M. *et al.* Three-dimensional structure of Escherichia coli glutathione S-transferase complexed with glutathione sulfonate: catalytic roles of Cys10 and His106. *J. Mol. Biol.* **281,** 135–147 (1998).
177. Datsenko, K. A. & Wanner, B. L. One-step inactivation of chromosomal genes in Escherichia coli K-12 using PCR products. *Proc. Natl. Acad. Sci. U. S. A.* **97,** 6640–6645 (2000).
178. Zwietering, M. H., Jongenburger, I., Rombouts, F. M. & van 't Riet, K. Modeling of the Bacterial Growth Curve. *Appl. Environ. Microbiol.* **56,** 1875–1881 (1990).
179. Carlin, A., Shi, W., Dey, S. & Rosen, B. P. The ars operon of Escherichia coli confers arsenical and antimonial resistance. *J. Bacteriol.* **177,** 981–986 (1995).
180. Vetting, M. W. *et al.* Crystal structure of a glutathione transferase family member from Salmonella enterica ty2, target efi-507262, with bound glutathione. *Be Publ.* null–null (2013).
181. Dan Kroll. A Visual Method for the Detection of Arsenic 0–500 µg/L. at <http://www.hach.com/asset-get.download.jsa?id=7639984489>
182. Scott, N., Hatlelid, K. M., MacKenzie, N. E. & Carter, D. E. Reactions of arsenic(III) and arsenic(V) species with glutathione. *Chem. Res. Toxicol.* **6,** 102–106 (1993).
183. Todorova T, V. S. Role of glutathione S-transferases and glutathione in arsenic and peroxide resistance in Saccharomyces cerevisiae: a reverse genetic analysis approach. *Biotechnol. Amp Biotechnol. Equip.* **21,** 348–352 (2007).
184. Zakharyan, R. A. *et al.* Human monomethylarsonic acid (MMA(V)) reductase is a member of the glutathione-S-transferase superfamily. *Chem. Res. Toxicol.* **14,** 1051–1057 (2001).
185. Board, P. G. *et al.* Identification, characterization, and crystal structure of the Omega class glutathione transferases. *J. Biol. Chem.* **275,** 24798–24806 (2000).
186. Denton, H., McGregor, J. C. & Coombs, G. H. Reduction of anti-leishmanial pentavalent antimonial drugs by a parasite-specific thiol-dependent reductase, TDR1. *Biochem. J.* **381,** 405–412 (2004).

187. Fyfe, P. K., Westrop, G. D., Silva, A. M., Coombs, G. H. & Hunter, W. N. Leishmania TDR1 structure, a unique trimeric glutathione transferase capable of deglutathionylation and antimonial prodrug activation. *Proc. Natl. Acad. Sci. U. S. A.* **109,** 11693–11698 (2012).
188. Tuomisto, H. A diversity of beta diversities: straightening up a concept gone awry. Part 1. Defining beta diversity as a function of alpha and gamma diversity. *Ecography* **33,** 2–22 (2010).
189. Shannon, C. E. A Mathematical Theory of Communication. *Bell Syst. Tech. J.* **27,** 379–423 (1948).